UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

Trabajo Fin de Grado

APLICACIÓN DE TÉCNICAS DE SOFT-COMPUTING A LA CLASIFICACIÓN AUTOMÁTICA DE GÉNEROS MUSICALES

Autor: Diego López Pajares
Director: Enrique Alexandre Cortizo

TRIBUNAL:

Presidente: Lucas Cuadra Rodríguez

Vocal 1º: Francisco Javier Acevedo Rodríguez

Vocal 2º: Enrique Alexandre Cortizo

CALIFICACIÓN:
FECHA:

A mi familia, que gracias a su apoyo incondicional
han hecho que este trabajo sea una realidad.

Como no podía ser de otra forma, aprovecho este documento para dar las gracias a personas que son muy importantes en mi día a día.

En primer lugar, me gustaría agradecerle al Dr. Enrique Alexandre todo su esfuerzo y de dedicación por explicarme con paciencia y detenimiento todas las dudas que me han surgido en estos meses. Además su simpatía y buena predisposición han hecho que en su despacho me sintiera como en casa.

También me gustaría mencionar a mis compañeros de clase que han hecho que estos años de universidad hayan sido inolvidables. Espero que sigamos llevándonos tan bien como hasta ahora siempre, y que nunca perdamos el contacto.

Las siguientes líneas van para mis amigos y amigas de toda la vida, esos con los que desde pequeños he pasado momentos únicos, y los que nos quedan por vivir. Las tardes jugando a tierra descubierta o al bote botero son épocas que no volverán, pero vendrán nuevos tiempos en los que nos los pasemos igual de bien, como por ejemplo en las fiestas del pueblo donde disfrutamos como auténticos enanos. Ojala que este buen rollo que tenemos dure para siempre.

Y como no, le tengo que agradecer a mi familia todo lo que han hecho por mí. Mención especial merecen mis padres, Juan y Milagros, que gracias al gran esfuerzo económico han hecho posible que termine mi carrera. Además sus consejos y como no, alguna que otra bronca han conseguido guiarme por el buen camino y han logrado que sea quien soy a día de hoy. También les agradezco que me hayan inculcado el valor del esfuerzo, haciéndome saber cuanto cuestan las cosas. En este apartado no podía olvidarme de mi abuelita, a la que quiero muchísimo. Ella es la que todos los días me saca una sonrisa y espero que lo siga haciendo durante muchos años más. Tampoco me olvido de mi hermano Samuel, que aunque a veces discutamos un poco sabe lo mucho que le quiero y todo lo que le digo siempre es por su bien. Y para finalizar también me acuerdo de mis tíos y primos que hacen que esas reuniones familiares sean unos de los mejores momentos de los cuales uno no se quiere desprender.

A todos, muchísimas gracias.

Índice General

Índice de figuras

Índice de Tablas

Resumen

El objetivo de este tfg es estudiar un método rápido y eficaz de clasificar archivos de audio según su género musical.

El trabajo partirá con la búsqueda de una base de datos ya etiquetada. Una vez que se ha descargado y adecuado la base de datos se extraen de ella características tímbricas con las que identifiquen a cada género musical. Una vez que se tienen las características se implementará un clasificador que se encargará de distinguir entre uno u otro género musical en función de las características de entrada. Por último se buscarán mejoras en función de los resultados obtenidos.

Palabras clave: base de datos**,** características, clasificador, género musical.

Abstract

The aim of this tfg is to study a fast and efficient method to classify audio files by genre.

The work will start with finding a database already labeled. Once downloaded the database, are extracted from it timbral features that identify each musical genre. Once you have the features, the classifier is responsible for classifying genres. Finally improvements will be sought based on the results obtained.

Keywords: database, features, classifier, musical genre.

Resumen extendido

La amplia variedad musical que existe en la actualidad y la complejidad que tiene la correcta clasificación por género musical han sido las razones del desarrollo de este trabajo fin de grado. Este capítulo describirá un amplio resumen de todo el proceso que se ha llevado a cabo para la realización de este proyecto.

El primer capítulo de esta memoria es una introducción del trabajo en la que se sitúan los antecedentes y se encuadra el presente trabajo.

El segundo capítulo trata sobre una base de datos musical etiquetada por géneros que servirá de base para los capítulos posteriores. Esta nueva sección incluye tres apartados: un primer apartado en el que se hace una introducción al capítulo, un segundo apartado en el que se explica los pasos que se han seguido hasta obtener la base de datos, y por último, un apartado en el que se explican las modificaciones que ha de sufrir la base de datos descargada con el objetivo de que sea válida para este proyecto.

El tercer capítulo habla de las características que posteriormente serán analizadas por el clasificador para poder clasificar correctamente todos los géneros musicales. El primer apartado que presenta este capítulo trata sobre el procesado de señal que se realiza a los archivos de audio, explicando por qué se decide entramar los archivos musicales. El siguiente apartado es una continuación del anterior en el que se explican todas las características que se desean recoger y el proceso que hay que seguir hasta conseguirlo. Por último se explica como se ha conseguido desarrollar todo lo anterior en el entorno de programación Matlab.

La cuarta sección trata sobre la herramienta que permite clasificar correctamente los géneros musicales propuestos. En un primer apartado se hará una introducción sobre las redes neuronales artificiales, las cuales intentan emular el comportamiento del aprendizaje humano. Para ello hacen uso de la generalización, el aprendizaje y la abstracción. La red SLFN “Single Layer Feedforward Network” pertenece al este tipo de redes y será la que te implemente en el proyecto. Por eso el siguiente apartado del proyecto trata sobre este tipo de redes, explicando su estructura y el modo de funcionamiento. Dentro de las redes SLFN, el algoritmo que se va a implementar es conocido como ELM “Extreme Learning Machine”. Este algoritmo es un algoritmo de

aprendizaje extremo que tiene la peculiaridad de ofrecer muy buenos resultados en muy poco tiempo. Se hará una descripción teórica del algoritmo explicando los pasos a seguir para ejecutarlo correctamente. Por último el capítulo recogerá como se ha logrado implementar este tipo de red en el entorno de programación Matlab.

En el siguiente capítulo por fin se monta completamente el clasificador musical. Se introducen las características extraídas en el capítulo 3 en los nodos de entrada de la red neuronal y se obtienen resultados. Se harán varios experimentos en los que se modifiquen parámetros importantes de la red neuronal con el objetivo de obtener unos resultados óptimos. Además en este capítulo se harán distintas normalizaciones a los parámetros de entrada de la red neuronal cuyo propósito es saber que normalización consigue los mejores resultados. Aun con todo esto los resultados que se obtienen no son nada halagüeños, por lo que el siguiente capítulo tratará de buscar una solución al problema.

El sexto capítulo busca una solución al problema anterior. Para ello se pensó en extraer más características de cada archivo de audio con el propósito de proporcionar información extra al clasificador. Con esta nueva información y la que ya se tenía del capítulo 3 se espera que el clasificador consiga identificar los cuatro géneros musicales propuestos con mayor facilidad. Serán los Coeficientes Cepstrales de Mel los encargados de aportar esa información extra que se necesita. Habrá un apartado en el que se explique con detalle la teoría acerca de los Coeficientes Cepstrales de Mel, posteriormente se explicará como se ha conseguido trasladar este proceso al entorno de programación Matlab. Por último se mostrarán los resultados que se obtienen a la salida de la red neuronal artificial cuando se introducen en la red las características del capítulo 3 más las características obtenidas en este capítulo. Los resultados mejoran pero no son los esperados, por lo que es necesario buscar nuevas soluciones que permitan solventar el problema. De ello se encargará el siguiente capítulo.

El séptimo capítulo introduce cambios en la red neuronal implementada hasta el momento. La novedad es que en vez de usar una única red que sea capaz de distinguir entre cuatro géneros musicales, se implementarán una serie de clasificadores binarios que sean capaces de distinguir mejor los géneros musicales propuestos. De esta forma es más fácil discernir entre los cuatro géneros musicales ya que cada clasificador solamente tendrá que identificar correctamente un género musical. Por ejemplo, la primera red neuronal tendrá que distinguir entre música clásica de la que no lo es, el segundo tendrá que determinar que archivos pertenecen al género etiquetado como música electrónica y descartar los que no lo sean. El proceso se repite hasta llegar al último clasificador. Esta tarea es más sencilla que la que tenía el clasificador multicanal implementado en los capítulos anteriores debido a que el clasificador multicanal tenía el

cometido de distinguir entre 4 géneros musicales distintos. Gracias a esta modificación se consiguen unos resultados realmente buenos que serán los resultados finales de este trabajo fin de grado.

El octava sección tiene como objetivo comprobar si los datos recogidos en el capítulo anterior son correctos. Para comprobarlo se extraerán las características explicadas anteriormente de una nueva base de datos y se introducirán en la red binaria de clasificadores. Este capítulo introduce una novedad, ahora se obtendrán dos resultados de la misma base de datos: un primer experimento en el que se introducen los datos en los clasificadores al igual que se ha hecho hasta ahora, y un nuevo experimento en el que los clasificadores simulan un comportamiento real.

El siguiente capítulo estará destinado a la dotación al proyecto de un entorno gráfico. Gracias a este entorno de programación será posible que un usuario común pueda hacer uso del clasificador musical. Se describirá el proceso que se ha seguido para crear la interfaz gráfica, así como su funcionamiento y utilidad.

Por último se presentarán unas conclusiones sobre la realización del proyecto, líneas futuras del trabajo, así como un pliego de condiciones y un presupuesto detallado del coste total del trabajo fin de grado.

1 Introducción

La música ha estado presente en la sociedad desde tiempos inmemoriales. Ya en la prehistoria los ritmos musicales gozaban de gran importancia para el hombre puesto que estos rudimentarios sonidos estaban presentes en los rituales de caza y en las fiestas, donde el hombre prehistórico bailaba sin descanso. Durante el paso del tiempo estos ritmos prehistóricos han ido evolucionando influenciados por factores sociales, culturales e históricos, dando lugar a la inmensa variedad musical que existe en la actualidad. Ante la existencia de una cantidad tan grande de música aparece la necesidad ordenarla adecuadamente para obtener una búsqueda eficiente. De esta necesidad surgen las distintas clasificaciones musicales que existen en la actualidad: género, autor, condición anímica, año,...

Dentro de los distintos tipos de clasificación, la clasificación por género musical es una de las más complicadas dado que existen numerosas imprecisiones que hacen que dos canciones de un mismo género musical contengan diferencias significativas en cuanto a características rítmicas e instrumentación. Estas imprecisiones han hecho que históricamente la clasificación por género musical haya sido realizada por expertos musicales, con la consecuente pérdida de tiempo y dinero que eso conlleva. Es por eso que la industria musical en la actualidad busque técnicas más eficientes de clasificación. Es aquí donde se encuadra este proyecto fin de grado, el cual tiene como objetivo mejorar el tiempo de ejecución y reducir el coste que conlleva clasificar el repertorio musical de la forma tradicional manteniendo un alto porcentaje de aciertos.

Para conseguir este objetivo se seguirán una serie de pasos bien estructurados. El primero de ellos es encontrar una base de datos musical que se encuentre etiquetada correctamente. Esta base de datos será el cimiento de este proyecto ya que sin la base de datos sería imposible poder desarrollar el trabajo. Posteriormente se trabajará sobre la base de datos adecuándola a las necesidades de este proyecto. Una vez que se ha adaptado esta base de datos se extraen de ella características musicales, las cuales servirán como patrones para que el clasificador identifique correctamente los géneros musicales propuestos. Una vez que se ha implementado el clasificador se obtienen resultados y se buscan mejoras que permitan mejorar los resultados iniciales. Con esas mejoras se consigue obtener un porcentaje de aciertos muy alto para todos los géneros musicales que servirán como resultados finales de este trabajo fin de grado.

2 Base de Datos Musical

2.1 Introducción

La obtención de una base de datos es un pilar básico para la realización de este proyecto. Gracias a esta base de datos es posible extraer información muy importante sobre cada género musical. Para conseguirlo se extraerán características musicales de todos los archivos de audio que contiene la base de datos y se asociarán con su correspondiente género musical. Los apartados siguientes mostrarán el proceso de búsqueda y de adaptación de la base de datos a las necesidades de este trabajo fin de grado.

2.2 Búsqueda de la base de datos

Internet es un medio que proporciona casi cualquier información si se utilizan las herramientas adecuadas. Como no podía ser de otra forma la base datos que se va a utilizar está dentro de esta gran red de información.

Desde el año 2005 se viene celebrando anualmente unas conferencias que investigan acerca de la información musical, más conocidas como MIREX, "Music Information Retrieval Evaluation eXange", Estás conferencias están gestionadas y coordinadas por el laboratorio de evaluación de sistemas de recuperación de información musical (IMIRSEL), y cada año en la web del MIREX se presentan los resultados de las investigaciones. Por todo esto, la página del MIREX es una buena opción para buscar una base de datos musical ya clasificada.

Dentro del apartado de clasificación por género musical de la página del MIREX 2005 (MIREX, 2005) se encuentran los enlaces de descarga de la base de datos que se ha utilizado (Music Technology Group, 2004). Esta base de datos viene dividida en dos partes, una primera parte llamada de entrenamiento la cual sirve para entrenar la red y comprobar los resultados, y una segunda parte llamada de validación que es la responsable de verificar que el porcentaje de aciertos que se ha obtenido al entrenar la red ha sido correcto.

La base de datos en su conjunto cuenta con 8 géneros musicales: clásica, electrónica, jazz, metal, punk, rock, pop y world. Por último cabe destacar que la base de datos de

entrenamiento cuenta con unas 800 canciones, mientras que la parte de validación está compuesta por 730 canciones aproximadamente.

2.3 Adecuación de la base de datos

Una vez que se ha descargado la base de datos, el siguiente paso es convertir la señal acústica a un formato numérico con el que se pueda trabajar. Para ello se va a convertir el formato audible a una serie de muestras que representan la envolvente de la señal musical. La herramienta que va a permitir realizar esta conversión y con la que se va a trabajar durante todo el proyecto es el programa de cálculo numérico Matlab.

El tipo de archivos del que está compuesta la base de datos es .mp3. Da la casualidad de que dicho formato no es compatible con Matlab, por lo que es necesario buscar una solución. Dos ideas son las que se han planteado en la búsqueda de la solución del problema:

- Convertir toda la base de datos a un formato que soporte Matlab (.wav) para posteriormente trabajar en Matlab con los archivos .wav.
- Buscar una librería de Matlab que logre convertir el formato .mp3 a un formato numérico. De esta forma es posible trabajar dentro del entorno de Matlab con el formato .mp3.

Al final se ha optado por usar la segunda opción por ser la solución más sencilla y que requiere menor tiempo de implementación.

La librería que se ha usado ha sido mp3read, obtenida en (Fernandez, 2004). Está compuesta por 2 funciones mp3read y mp3write y un programa externo. Lo que hace la librería básicamente es convertir a .wav el archivo .mp3 mediante el programa externo que incorpora, para que Matlab pueda trabajar con un formato soportado.

De las dos funciones que trae la librería, la que se va a emplear es mp3read. Esta función proporciona la señal de audio muestreada y cuyos valores de envolvente oscilan en el rango [-1,1], así como la frecuencia de muestreo, el número de bits por muestra y datos sobre la codificación y etiquetas del archivo. Una vez que la función devuelve la señal muestreada es necesario convertirla a formato monofónico. De esta forma se facilita el cálculo de la extracción de características de las canciones que componen la base de datos. Para la conversión a formato monofónico se ha optado por sumar los dos canales de audio y dividir el resultado entre dos. Una vez que se ha hecho esto, la señal muestreada está lista para procesar y poder extraer de ella las características más relevantes.

En cuanto a los géneros musicales que tiene la base de datos, cabe decir que no todos los géneros tienen el mismo número de canciones. Este hecho hace que a la hora de clasificar el clasificador de más importancia al género con más archivos, por lo que se tomó la decisión de reorganizar los géneros musicales que tiene la base de datos. A la hora de reorganizar la base de datos se optó por eliminar algún género musical, así como reagrupar otros. La Tabla 1muestra el antes y el después de los géneros musicales en la base de datos:

Base datos original	
Género	**Nº canciones**
Classical	318
Jazz	26
Electronic	115
Pop	6
Metal	29
Rock	95
Punk	16
World	122

Nueva base datos	
Género	**Nº canciones**
Classical	318
Electronic	115
Metal/Rock/Punk	140
World	122

Tabla 1: Comparativa entre los géneros de la base de datos original y la base de datos modificada.

Como se puede apreciar en la Tabla 1, la base de datos original contenía 8 géneros musicales, de los cuales algunos apenas tenían asociados archivos de audio. Para compensar los géneros de prescindió de jazz y pop, y se agruparon en un mismo género metal rock y punk por su afinidad musical. De esta forma se consigue equilibrar el número de archivos que contiene cada género musical facilitando la tarea al clasificador que se implementará posteriormente.

3 Procesado de señal y extracción de características

Este capítulo será el encargado de adquirir información muy relevante sobre cada uno de los géneros musicales. Con la aplicación de diversas técnicas de análisis y procesado de señal se conseguirá obtener 10 características de cada fichero de audio.

La estructura de este capítulo consta de 3 apartados: un primer apartado en el que se entraman las señales acústicas, un segundo apartado en el que se obtienen las características tímbricas de las señales acústicas, y por último, un apartado dedicado a la programación de todas las ecuaciones necesarias para la obtención de características en el entorno de desarrollo Matlab.

Toda la documentación necesaria para poder desarrollar este capítulo se ha obtenido de (Tzanetakis & Cook, 2002), (Nam, 2001) y (Alvarado, 2005).

3.1 Procesado de señal

Antes de extraer las características es necesario procesar la información para poder obtener mejores resultados. A la hora de procesar los datos se ha tomado la determinación de entramar cada señal de audio en tramas cuya longitud asciende a 2048 muestras. El número de muestras no se ha escogido al azar, sino que se fundamenta en la naturaleza aleatoria y no estacionaria que poseen las señales acústicas. La aleatoriedad y la naturaleza no estacionaria de las señales acústicas suponen un gran inconveniente a la hora de obtener las características espectrales, y para solucionarlo, se recurre a utilizar ventanas de muy corta duración. Es aquí donde adquiere una importancia vital la longitud de las tramas: La duración de trama que se está manejando es de 46.3 ms (frecuencia de muestreo de 44.1 kHz en una ventana de 2048 muestras). Dada la corta duración de la trama se consigue que la señal que contiene cada trama sea prácticamente estacionaria, facilitando así la posterior extracción de características. Además gracias al entramado se consigue obtener más información puesto que se calculará la media y la desviación típica de todas las tramas que componen una característica, obteniéndose así dos medidas en vez de una. Por ejemplo, se quiere calcular la energía de una señal acústica. Para ello se calculará la energía de todas las tramas que componen la señal acústica. Posteriormente se calculará la media y la desviación típica de las energías calculadas. Así se obtienen dos medidas

de una misma característica en vez de una sola que se obtendría si se calculase la energía de la señal directamente sin entramar.

3.2 Extracción de características

Este apartado explicará de manera teórica las técnicas que se van a emplear para la extracción de las características más relevantes de una señal acústica.

La extracción de características es un proceso de cálculo que permite caracterizar a una señal acústica. El proyecto se centrará esencialmente en las características musicales relacionadas con el timbre, pudiendo separarse en dos tipos: características temporales y espectrales. A continuación se analizarán con detalle cada tipo.

- Características temporales:

  Dentro de este grupo se encuentran la energía de la señal acústica y el número de cruces por cero, más conocido como zerocrossing. Ahora se describirá con más detalle las ecuaciones y el proceso que se ha seguido para la obtención de estos dos parámetros.

  Energía de la señal: se calcula la energía de cada trama como el módulo de la trama al cuadrado según la ecuación (1).

$$E_k = \sum_{n=0}^{N} |x[n]|^2 \qquad (1)$$

  Con la energía de cada trama se obtiene la energía media y la desviación típica de la energía de la señal. Además en el cálculo de este parámetro se ha introducido una mejora que elimina las partes de audio que no contienen sonidos. Para ello se ha establecido un umbral que elimina las tramas cuya energía no supere ese umbral.

  Zerocrossing: este parámetro da una idea de lo ruidosa que es la señal puesto que cuenta el número de pasos por cero de cada trama, cuanto mayor es el número de pasos por cero, mayor será el ruido existente. Además este es uno de los parámetros que permite discriminar entre la voz y el audio. Para el cálculo del número de pasos por cero en cada trama se ha hecho uso de la ecuación (2)

$$Z_k = \frac{1}{2} \cdot \sum_{n=0}^{N-1} |signo(x[n]) - signo(x[n+1])| \quad (2)$$

Posteriormente se obtiene la desviación típica y la media de los pasos por cero de todas las tramas.

- Características espectrales:

  A la hora de obtener las características espectrales se ha decidido enventanar todas las tramas con la ventana de Hamming. Una de las razones por las que se ha decidido utilizar esta ventana es el suavizado que logra en los bordes de la trama, evitando así las discontinuidades bruscas entre tramas. La lógica dice que utilizando una ventana rectangular no se modificaría la señal en el dominio del tiempo por lo que las características espectrales se verían menos afectadas, pero en realidad esto no es así. De aquí se obtiene la segunda razón por la cual se utiliza la ventana de Hamming. Una multiplicación en el dominio temporal se convierte en una convolución en el dominio frecuencial, por eso para las características espectrales es necesario utilizar la ventana que más se aproxime a la función delta, puesto que una convolución con una delta no modifica el espectro de la señal original. Analizando ambas ventanas se puede observar que el espectro de la ventana de Hamming es la que más se aproxima a la función delta como muestra la Figura 1.

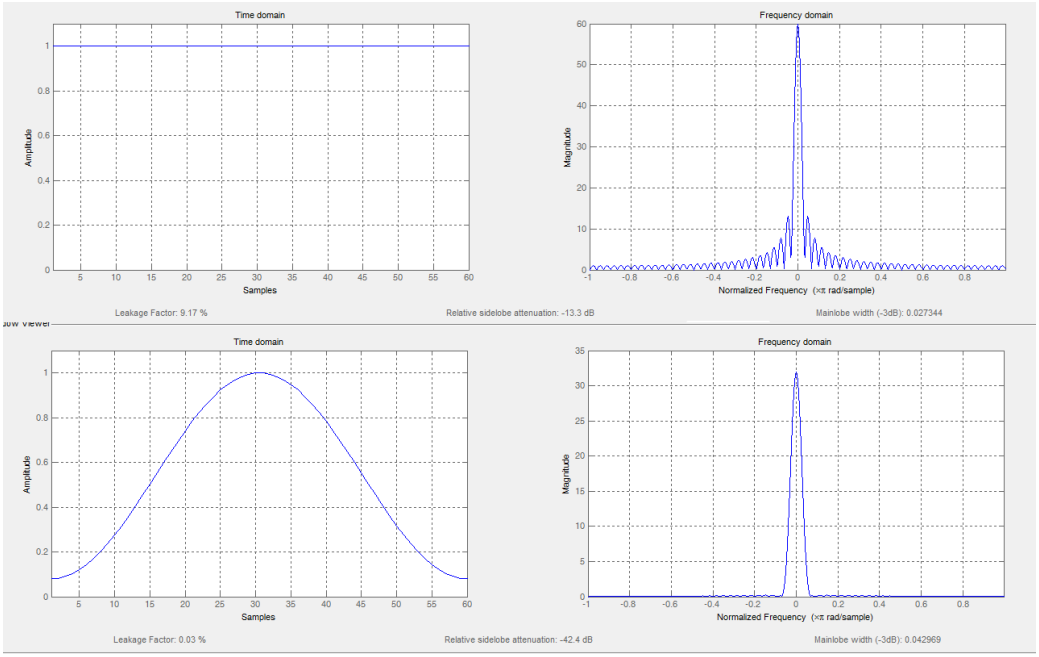

Figura 1 Tipos de ventanas y su espectro

Con el tipo de ventana ya escogido, el siguiente paso es empezar a extraer las características deseadas, las cuales se detallan a continuación:

Centroide espectral: está definido como el centro de gravedad del espectro, es decir, la frecuencia que divide al espectro en dos partes iguales. Este parámetro está relacionado con la medida del "brillo" del sonido. De esta forma se relaciona a los sonidos más "brillantes" con los valores de centroide más altos, los cuales están compuestos de frecuencias altas. Para el cálculo del centroide en cada trama se hace uso de la ecuación (3)

$$C_k = \frac{\sum_{n=0}^{N} M_k[n] \cdot n}{\sum_{n=0}^{N} M_k[n])} \tag{3}$$

Dónde $M_k[n]$ es la amplitud de la transformada de Fourier a la frecuencia n de la trama k. Una vez calculado el centroide de todas las tramas se obtiene la media y la desviación típica de todos los centroides.

Factor de Rolloff: se define como la frecuencia R por debajo de la cual se concentra el 85% del módulo del espectro. Este parámetro da una idea de la forma que puede adoptar el espectro de la señal. Para el cálculo de este parámetro en cada trama se hace uso de la ecuación (4)

$$\sum_{n=0}^{R} M_k[n] = 0.85 \cdot \sum_{n=0}^{N} M_k[n] \tag{4}$$

Al igual que en el caso anterior, se calcula la media y la desviación típica del factor de rolloff con el factor de rolloff de cada trama.

Flujo espectral: es la diferencia al cuadrado del módulo del espectro entre dos tramas consecutivas. Esta medida indica la rapidez con la que cambia la energía del espectro. La ecuación (5) es la que se ha utilizado para el cálculo del flujo espectral.

$$F_k = \sum_{n=0}^{N} (M_k[n] - M_{k-1}[n])^2 \tag{5}$$

Donde $M_k[n]$ y $M_{k-1}[n]$ son el módulo de la transformada de Fourier de la trama k y de la trama anterior respectivamente.

Finalmente se ha obtenido la media y la desviación típica del flujo espectral para todas las tramas.

3.3 Implementación en Matlab

Para obtener las características explicadas en el apartado anterior se ha optado por programar unos scripts en el entorno de programación Matlab. Se ha creado un script para cada característica siguiendo las ecuaciones descritas en el apartado anterior. Una vez que se implementaron todos estos scripts se creó otro que permite organizar y almacenar todas estas características. El formato utilizado para su almacenaje consiste en un matriz de 11 columnas y tantas filas como archivos de audio contiene la base de datos. Cada fila tiene una primera columna con la etiqueta correspondiente al género

musical al que pertenece el archivo y 10 columnas más en las que están las características extraídas. A modo de ejemplo la Tabla 2 muestra la organización de esta matriz de características:

Etiqueta	Media centroide	Desviación típica centroide	Media Roloff	Desv. típica Roloff	Media flujo espectral	Desv. típica flujo espectral	Media zerocrossing	Desv. típica zerocrossing	Media energía	Desv. típica energía
1	0,006	0,01	1023,85	26,19	127,48	100,0	0,04	0,12	0,02	0,01
2	0,012	0,011	1021,9	14,8	260,6	185,0	0,177	0,287	0,064	0,069
3	0,035	0,017	1024,3	11,11	243,67	52,14	0,32	0,15	0,085	0,02
4	0,015	0,011	1023,5	21,71	233,7	144,0	0,138	0,116	0,035	0,023

Tabla 2: Estructura de la matriz que almacena las características musicales. Los datos que aparecen en la tabla se corresponden con ejemplos reales del TFG.

El campo etiqueta posee valores que van del 1 al 4 cuya correspondencia con un género musical se muestra en la Tabla 3.

Valor numérico	**Género musical correspondiente**
1	Classical
2	Electronic
3	Metal/Rock/Punk
4	World

Tabla 3: Correspondencia entre la etiqueta de la matriz de características y su género musical

Una vez que se ha construido la matriz completamente se procede a guardar el archivo en el disco duro del ordenador en el formato .mat soportado por Matlab para poder trabajar posteriormente con esta matriz de características.

4 Clasificación

4.1 Introducción

Para la clasificación de los distintos géneros musicales se va a implementar una red neuronal artificial. Este tipo de redes intentan emular el procesamiento de información de sistemas nerviosos humanos, es por eso que las redes neuronales artificiales presentan características propias del cerebro humano:

Aprendizaje: El conocimiento de las cosas lo obtienen a través del estudio, ejercicio o experiencia. Este hecho hace que las redes neuronales artificiales posean la capacidad de cambiar dinámicamente según las condiciones del medio.

Generalización: Sacan conclusiones globales a partir de casos particulares. Gracias a esta característica estas redes pueden conseguir resultados correctos a entradas que están afectadas por ruido o distorsiones siempre que estén dentro de un margen.

Abstracción: Separan las cualidades de un objeto para considerarlas aisladamente. Esto permite abstraer un conjunto de datos de entrada que no muestran características comunes.

Las cualidades descritas anteriormente hacen que la red tenga un comportamiento no lineal, lo que permite procesar información de fenómenos no lineales. Este hecho es de gran ayuda dado que la mayoría de los sucesos presentes en la naturaleza carecen de linealidad.

El tipo de red que se va a emplear es conocida como "Single Layer Feed-forward Network" (SLFN), cuya descripción y análisis en profundidad se verá con mayor detalle en el siguiente apartado.

4.2 Redes SLFN

Las redes SLFN constan de una estructura dividida en sustratos o capas. Cada capa tiene una serie de neuronas interconectadas que permiten aproximar cualquier función continua si se seleccionan bien los híper-parámetros de la red.

El esqueleto de una red SLFN básica consta de 3 estratos, los cuales se describen a continuación:

- Capa de entrada: en esta capa se sitúan las neuronas que contienen los patrones de entrada a la red.

- Capa intermedia: conocida como capa oculta. Aquí se encuentran neuronas cuya entrada proviene de la capa anterior (capa de entrada) y cuya salida pasa a la capa posterior (capa de salida).

- Capa de salida: en esta última capa se encuentran las neuronas que contienen las salidas de la red neuronal.

La Figura 2 muestra gráficamente la estructura de la red.

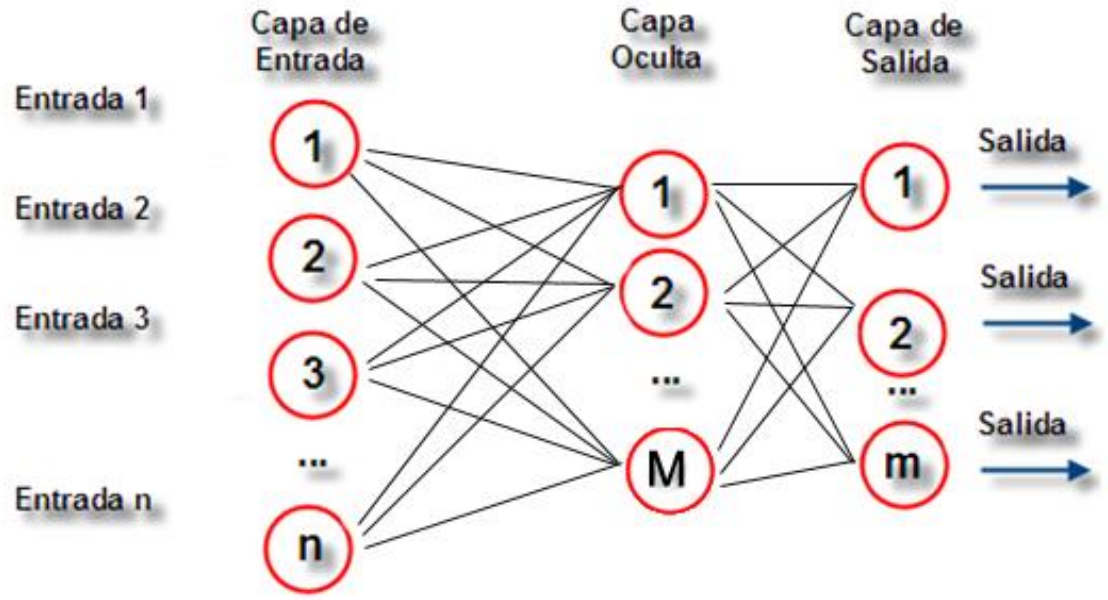

Figura 2: Estructura de una red neuronal.

Entrando más en detalle, una red SLFN típica está compuesta por una capa oculta de M neuronas. Los pesos de la capa de entrada conectan las n variables de entrada con las M neuronas. A continuación una nueva capa de pesos vuelven a conectar la salidas de las M neuronas de la capa oculta con las m salidas de la red.

Se considera un vector de entrada con N muestras aleatorias $x_j = [x_{j1}, x_{j2},..., x_{jn}]^T \in \Re^n$, cuyas soluciones son $t_j = [t_{j1}, t_{j2}, .., t_{jm}]^T \in \Re^m$. El resultado de cada neurona de salida viene dado por la expresión (6)

$$o_j = \sum_{i=1}^{M} \beta_i \cdot f(w_i \cdot x_j + b_i) \ , \quad j = 1 \dots N \tag{6}$$

Donde $f(\cdot)$ son las funciones de activación de las neuronas de la capa oculta; $\beta_j=[\beta_{j1}, \beta_{j2}, \dots, \beta_{jm}]^T$ son los pesos de salida de la neurona oculta j-ésima; $w_j=[w_{j1}, w_{j2}, \dots, w_{jn}]^T$ es el vector de pesos que conecta la j-ésima neurona oculta con las características de entrada y b_j es el sesgo de la j-esima neurona oculta. Además $w_j \cdot x$ es el producto escalar de w_j y x.

Las funciones de activación típicas de los nodos ocultos suelen ser del tipo sigmoide, en ese caso la SFLN se conoce como MLP ("Multi Layer Perceptron"). También es posible utilizar otras funciones de activación como las de base radial, pasando a llamarse la red en este caso como RBF ("Radial Basis Function"). Por el contrario las neuronas de salida tienen la ventaja de tener funciones de activación lineales.

Por otra parte, los híper-parámetros que permiten modificar y ajustar la SLFN se corresponden con los pesos de la red y el número de nodos. Un ajuste óptimo de estos parámetros permite que a la salida se obtengan resultados con un alto porcentaje de aciertos.

A la hora de implementar este tipo de redes, lo más habitual es fijar un número de neuronas M e inicializar los pesos aleatoriamente para posteriormente entrenar la red con un conjunto de entrenamiento y con el algoritmo "back propagation" junto con métodos de optimización basados en gradiente. Mediante validación cruzada se obtiene el número óptimo de neuronas. Este procedimiento supone un alto coste computacional debido al elevado número de pasos que se requieren para entrenar la red y a la búsqueda del número de neuronas.

Debido a la lentitud de este algoritmo han surgido mejoras que permiten entrenar la red mejorando el tiempo de ejecución y manteniendo la precisión en los aciertos. El algoritmo que consigue lo descrito anteriormente se denomina ELM (Extreme learning machine), y es el algoritmo que se va a emplear en este trabajo. En el siguiente apartado se detallará con más precisión el algoritmo ELM.

4.3 ELM: Extreme Learning Machine.

Este algoritmo se basa en que una SLFN compuesta por M neuronas, cuyos pesos de entrada se inicializan aleatoriamente, puede aprender N casos de entrenamiento sin producir errores, siendo $N \geq M$. Al asignar estos pesos al azar, la SLFN puede considerarse como un sistema lineal pudiéndose determinar los pesos de salida mediante la inversa generalizada.

Como ya se había visto anteriormente, una SLFN con N muestras de entrada aleatorias a la salida de cada neurona produce resultados según la expresión (6).

En general, una SLFN con M neuronas en la capa oculta puede aproximar estas N muestras con error 0:

$$\sum_{j=1}^{M} \|o_j - t_j\| = 0 \tag{7}$$

Para que se cumpla la expresión (7) deben existir w_i, β_i y b_i tal que:

$$\sum_{i=1}^{M} \beta_i \cdot f(w_i \cdot x_j + b_i) = t_j, \quad j = 1 \dots N \tag{8}$$

Las anteriores N ecuaciones de forma compacta se pueden escribir como:

$$HB = T \tag{9}$$

donde

$$H(w_1, \dots, w_M, b_1, \dots, b_m, x_1, \dots x_N) = \begin{bmatrix} f(w_1 \cdot x_1 + b_1) & \dots & f(w_M \cdot x_1 + b_M) \\ \vdots & \dots & \vdots \\ f(w_1 \cdot x_N + b_1) & \dots & f(w_M \cdot x_N + b_1) \end{bmatrix} \tag{10}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{Mxm} \tag{11}$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{Nxm} \tag{12}$$

Donde H es la matriz de salida de la capa oculta de la SLFN, y la columna i-ésima se corresponde a la salida de la neurona i-ésima de la capa oculta con respecto a las salidas $x_1,\ldots, x_N$.

Para poder desarrollar el algoritmo ELM Huang et al. desarrollan en (Guangbin , Quin-Yu, & Chee-Kheong , Extreme learning machine: Theory and aplications, 2006) una serie de teoremas. Con ayuda de estos teoremas se llega a la conclusión de que si se fijan los valores de w_i y b_i el entrenamiento de una red SLFN es equivalente a encontrar una solución de mínimos cuadrados $\hat{B}$ del sistema lineal HB=T:

$$\hat{B} = \arg\min\|H(w_1, w_M, b_1, \ldots b_M)B - T\| \tag{13}$$

Si M=N (número de muestras es igual al número de neuronas ocultas) la matriz H es cuadrada e invertible pudiendo aproximar la SLFN a error cero las muestras de entrenamiento. Sin embargo, en la mayoría de los casos esto no ocurre ya que el número de nodos de la capa oculta suele ser menor que el número de vectores de entrenamiento. En ese caso la matriz no sería cuadrada, por lo que no existiría β_i que cumpliera HB=T. En ese caso y con ayuda del teorema 5.1 propuesto en el apéndice del artículo (Guangbin , Quin-Yu, & Chee-Kheong , Extreme learning machine: Theory and aplications, 2006) se llega a la conclusión de que la solución con menor norma de mínimos cuadrados del sistema anterior es:

$$\hat{B} = H^{+}T \tag{14}$$

Donde H^{+} es la inversa de la matriz generalizada de Moore-Penrose.

De esta forma ELM es capaz de proporcionar un entrenamiento rápido y eficiente para una red SLFN, aunque es necesario establecer el número de nodos de la capa oculta.

A modo de resumen de detalla a continuación el algoritmo ELM:

Dado un conjunto de entrenamiento $E=\{(x_j,t_j),\ x_j \in \Re^n,\ t_j \in \Re^m,\ j=1,\ldots,N\}$, un función de activación $f(\cdot)$ y un número de neuronas en la capa oculta M,

1) Se asignan aleatoriamente los pesos de la capa de entrada w_i y sesgos b_i, i=1,…, N.

2) Se calcula la matriz de salida de la capa oculta usando la ecuación (10).

3) Se calcula la matriz de pesos de la capa de salida B=H^{+}T, donde B y T están definidos en (11)y (12) respectivamente, y siendo H^{+} la matriz inversa generalizada de Moore-Penrose de H.

Para documentar toda la parte teórica se ha hecho uso de las siguientes referencias bibliográficas: (Guangbin , Quin-Yu, & Chee-Kheong , Extreme learning machine: Theory and aplications, 2006), (Guangbin, Hongming, Xiaojian, & Rui, 2012), (Crespo, 2013) y (García Laencina, Verdú Monedero, Larrey Ruiz, Morales Sánchez, & Sancho Gómez, 2010).

4.4 Implementación en Matlab

A la hora de usar la ELM en Matlab se optó por descargar una librería con la ELM ya programada. La página de descarga (Guangbin, Extreme Learning Machine) posee teoría sobre las ELM y librerías con la ELM implementada para varios lenguajes de programación así como ramales de la ELM original modificados para diversas soluciones. Dentro de la librería que contiene la ELM básica existen 3 funciones, una función básica que entrena la red y predice resultados, y otras dos "avanzadas" que permiten entrenar la red y predecir los resultados por separado. En este caso se van a utilizar la funciones "avanzadas" cuyo enlace de descarga es (Guangbin, Extreme Learning Machine).

La librería que se va a utilizar posee dos funciones, elm_train y elm_predict, las cuales se van a explicar con un poco más de detalle a continuación:

- Elm_train es la función que permite entrenar la red neuronal. La Tabla 4 muestra el prototipo de la función y explica los argumentos de entrada y de salida de dicha función.

<table>
<tr><td colspan="3">[TrainingTime,TrainingAccuracy] = elm_train(TrainingData_File, Elm_Type, NumberofHiddenNeurons, ActivationFunction)</td></tr>
<tr><td colspan="3">Argumentos de entrada</td></tr>
<tr><td>TrainingData_File</td><td colspan="2">Base de datos con los archivos de entrenamiento. En la primera columna de cada fila debe estar el</td></tr>
<tr><td rowspan="2">Elm_Type</td><td rowspan="2">Tipo de ELM</td><td>0- Regresión</td></tr>
<tr><td>1- Clasificación</td></tr>
<tr><td>NumberofHiddenNeurons</td><td colspan="2">Número de neuronas que tendrá la capa oculta de la red.</td></tr>
<tr><td rowspan="3">ActivationFunction</td><td rowspan="3">Función de activación</td><td>Función Sigmoidal</td></tr>
<tr><td>Función Sine</td></tr>
<tr><td>Función Hardlim</td></tr>
<tr><td colspan="3">Argumentos de salida</td></tr>
<tr><td>TrainingTime</td><td colspan="2">Tiempo que tarda en entrenarse la red</td></tr>
<tr><td>TrainingAccuracy</td><td colspan="2">Porcentaje de aciertos en la fase de entrenamiento.</td></tr>
</table>

Tabla 4: Prototipo de la función de entrenamiento y sus parámetros.

- Elm_predict: es la función que da la solución a unas características de entrada una vez que la red neuronal está entrenada. El prototipo de la función y sus argumentos de detallan en la Tabla 5.

[TestingTime, TestingAccuracy] = elm_predict(TestingData_File)	
Argumentos de entrada	
TestingData_File	Datos de entrada a partir de los cuales se va a predecir el resultado de la salida.
Argumentos de salida	
TestingTime	Tiempo que tarda la red en predecir el resultado.
Testing Accuracy	Porcentaje de aciertos que se obtienen a la salida de la red neuronal.

Tabla 5: Prototipo de la función de predicción de resultados y sus argumentos.

El proceso a seguir para la implementación de la ELM es entrenar la red con la función elm_train escogiendo un número de nodos al azar y una función de activación para posteriormente obtener resultados con elm_predict. Si los resultados no se ajustan a lo buscado se cambian el número de nodos y la función de activación hasta que se obtengan unos resultados aceptables.

5 Primera aproximación

En este nuevo capítulo se van a presentar los resultados que se obtienen una vez que se ha extraído de la base de datos descargada las características de capítulo 3 y se ha implementado el clasificador del capítulo 4. Los resultados que se van a presentar se estructurarán en 3 apartados: el primero presentará los resultados que proporciona el clasificador si se introducen los datos de la base de datos sin normalizar, el segundo con la base de datos normalizada por el valor máximo y el tercero con la base de datos normalizada en media cero y varianza uno. Con esto se pretende observar cual de los 3 métodos obtiene mejores resultados.

El clasificador implementado en todos los casos es una ELM funcionando en modo clasificación multiclase con 4 nodos de salida, cada salida se corresponde con uno de los 4 géneros musicales.

5.1 Experimento 1: Base de datos sin normalizar

Como se ha explicado anteriormente, en este experimento se introdujo la base de datos sin normalizar en el clasificador multicanal. Los datos que se muestran a continuación proporcionan el porcentaje de aciertos que ha tenido cada género musical, así como las matrices de confusión entre los distintos géneros musicales.

En un primer momento se presentará el porcentaje de aciertos para cada género teniendo en cuenta el número de nodos y la función de activación empleada. Las matrices de confusión solamente se mostrarán en los casos con mejores resultados. Para estos experimentos el número de características que se tienen en cuenta son las explicadas en el capítulo 3 (10 características en total).

La Tabla 6 muestra el porcentaje de aciertos para un número distintos de la capa oculta si la función de activación es sigmoidal.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	100,00	0,00	0,00	0,00	25,00
20	100,00	0,00	0,00	2,70	25,68
100	92,71	22,86	78,57	0,00	48,54
500	89,58	45,71	69,05	21,62	56,49

Tabla 6: Resultados que proporciona el clasificador con una función de activación sigmoidal y distinto número de nodos en la capa oculta.

La Tabla 7 recoge los resultados que proporciona el clasificador para distinto número de nodos en la capa oculta y función de activación sine.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	27,08	20	16,67	16,22	19,99
20	41,67	22,86	9,52	21,62	23,92
100	29,17	8,57	19,05	29,73	21,63
500	33,33	17,14	30,95	21,62	25,76

Tabla 7: Porcentaje de aciertos por género con función de activación sine y distinto número de nodos en la capa oculta.

Por último la Tabla 8 muestra los resultados de salida del clasificador cuando la función de activación empleada es una función hardlim.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	27,08	20	16,67	16,22	19,99
20	41,67	22,86	9,52	21,62	23,92
100	29,17	8,57	19,05	29,73	21,63
500	87,5	40	90,47	18,91	59,22

Tabla 8: Porcentaje de aciertos por género musical cuando la función de activación empleada es una función hardlim.

Observando los resultados de las tres tablas anteriores se ve que el porcentaje de aciertos obtenidos es pésimo, ya que en ningún caso se obtiene una tasa de aciertos mayor del 50% en todos los géneros musicales, sino que algún género obtienes resultados aceptables mientras otros los resultados que presentan son muy malos. Estos hechos hacen que la media global en el mejor de los casos ronde el 60% de aciertos.

A continuación las tablas Tabla 9,Tabla 10 yTabla 11 muestran las matrices de confusión de los casos que tienen la mayor media global de aciertos para cada función de activación. Da la casualidad que en todas las funciones de activación los mejores resultados se obtienen con la configuración de 500 nodos.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	89,58%	0,00%	8,33%	2,08%
Electronic	17,14%	45,71%	34,29%	2,86%
Metal/Rock/Punk	9,52%	21,43%	69,05%	0,00%
World	35,14%	24,32%	18,92%	21,62%

Tabla 9: Matriz de confusión para función de activación sigmoidal y 500 nodos en la capa oculta.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	33,33%	25,00%	25,00%	16,67%
Electronic	37,14%	17,14%	25,71%	20,00%
Metal/Rock/Punk	19,05%	21,43%	30,95%	28,57%
World	40,54%	13,51%	24,32%	21,62%

Tabla 10: Matriz de confusión para función de activación sine y 500 nodos en la capa oculta.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	87,50%	0,00%	7,29%	5,21%
Electronic	17,14%	40,00%	40,00%	2,86%
Metal/Rock/Punk	4,76%	4,76%	90,48%	0,00%
World	40,54%	13,51%	27,03%	18,92%

Tabla 11: Matriz de confusión para función de activación hardlim y 500 nodos en la capa oculta.

De las 3 funciones de activación las que mejores resultados dan a la hora de clasificar correctamente los géneros musicales son la función sigmoidal y la hardlim, ambas con una configuración de 500 nodos en la capa oculta, ya que la media global de aciertos que llegan a obtener está cerna al 60%. Este 60% no es del todo real, ya que existen oscilaciones muy amplias entre la media de aciertos de los géneros musicales. Por ejemplo, en los géneros musicales etiquetados como classical y metal/rock/punk se obtienen tasas de acierto muy aceptables (mayores del 70%) frente a tasas inferiores al 50% que presentan los géneros etiquetados como electronic y world.

En cuanto a las matrices de confusión, las tablas Tabla 9, Tabla 10 y Tabla 11 muestran el porcentaje de aciertos y fallos de cada género musical. Las tablas Tabla 9 y Tabla 11 son las que representan las mejores tasas de aciertos en el global, y por tanto serán las tablas en las que centrarse. Ambas tablas presentan un porcentaje de aciertos muy bueno en los géneros classical y metal/rock/punk. Los errores que presentan estos dos géneros se deben a la confusión del género classical con los géneros metal/rock/punk y world con una tasa de fallos inferior al 10% en cada uno de los dos géneros, mientras que el género metal/rock/punk tiende a confundirse con los géneros electronic y classical, siendo más propenso para la confusión el primero de ellos (en la Tabla 9 se etiquetan el 21% de las canciones metal/rock/punk como electronic, frente

al 9% que son etiquetadas como classical). Por el contrario los dos géneros musicales restantes (electronic y world) presentan un bajo porcentaje de aciertos llegando a confundir hasta el 40% de las canciones con un género musical diferente.

Estos resultados son inadmisibles para un sistema de reconocimiento de géneros musicales.

Los malos datos pueden ser debidos a que la base de datos está sin normalizar, puesto que en el manual de la ELM implementada aconseja que todas las características que se van a utilizar para crear patrones de reconocimiento deberían estar normalizadas y acotadas dentro del rango [-1,1].

En los siguientes experimentos se comprobará si esta tónica continúa o si por el contrario remite gracias a la normalización. El objetivo a conseguir es obtener altos porcentajes de aciertos en todos los géneros musicales. De no ser así se tomarán medidas que intenten solucionar el problema.

5.2 Experimento 2: Base de datos normalizada por el máximo

En este nuevo experimento todas las características que están en la base de datos se van a normalizar por su valor máximo, así se consigue que todas las características de la base de datos estén comprendidas dentro del rango [-1,1], como indica la recomendación de la ELM. Al igual que en el experimento anterior, 10 van a ser las características que permitan clasificar los 4 géneros musicales correctamente.

El procedimiento a seguir va a ser el mismo que se ha seguido en el experimento anterior. Primero se mostrarán las matrices que muestran el porcentaje de acierto de cada género musical para distinto número de nodos y funciones de activación, y luego se expondrán las matrices de confusión para los mejores resultados.

Siguiendo los pasos descritos en el párrafo anterior, el primer paso es mostrar los resultados para distintos nodos y funciones de activación. De ello se encargarán las tablas Tabla 12, Tabla 13 y Tabla 14:

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	66,66	14,28	90,47	2,7	43,53
20	73,95	45,71	88,09	2,7	52,61
100	89,58	45,71	90,47	8,1	58,47
500	41,66	11,42	40,47	35,13	32,17

Tabla 12: Porcentaje de aciertos por género con función de activación sigmoidal y distinto número de nodos en la capa oculta.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	86,45	2,85	85,71	18,91	48,48
20	73,95	11,428	78,57	16,21	45,04
100	86,45	45,71	76,19	18,91	56,82
500	15,62	17,14	40,47	40,54	28,44

Tabla 13: Porcentaje de aciertos por género con función de activación sine y distinto número de nodos en la capa oculta.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	71,87	0,00	92,85	2,70	41,86
20	54,16	57,14	45,23	5,40	40,48
100	68,75	31,42	83,33	24,32	51,96
500	26,04	54,28	47,61	27,02	38,74

Tabla 14: Porcentaje de aciertos por género con función de activación hardlim y distinto número de nodos en la capa oculta.

Las siguientes tablas reflejan las matrices de confusión para los mejores casos de cada función de activación. Da la casualidad de que el mejor caso se ha dado para 100 nodos.

	Classical	**Electronic**	**Metal/Rock/Punk**	**World**
Classical	89,58%	0,00%	8,33%	2,08%
Electronic	14,29%	45,71%	37,14%	2,86%
Metal/Rock/Punk	2,38%	7,14%	90,48%	0,00%
World	32,43%	16,22%	43,24%	8,11%

Tabla 15: Matriz de confusión para 100 nodos en la capa oculta y función de activación sigmoidal.

	Classical	**Electronic**	**Metal/Rock/Punk**	**World**
Classical	86,46%	0,00%	11,46%	2,08%
Electronic	14,29%	45,71%	37,14%	2,86%
Metal/Rock/Punk	4,76%	9,52%	76,19%	9,52%
World	29,73%	27,03%	24,32%	18,92%

Tabla 16: Matriz de confusión para 100 nodos en la capa oculta y función de activación sine.

	Classical	**Electronic**	**Metal/Rock/Punk**	**World**
Classical	68,75%	3,13%	17,71%	10,42%
Electronic	11,43%	31,43%	57,14%	0,00%
Metal/Rock/Punk	4,76%	4,76%	83,33%	7,14%
World	24,32%	18,92%	32,43%	24,32%

Tabla 17: Matriz de confusión para 100 nodos en la capa oculta y función de activación hardlim.

En este nuevo caso de estudio no se han conseguido mejorar los resultados obtenidos en el experimento anterior puesto que en este nuevo experimento la mejor media global de

aciertos asciende al 58,47%(función activación sine y 100 nodos), frente al 59.22% que se obtuvo en el experimento 1. Viendo las media globales de aciertos se puede decir que los resultados prácticamente idénticos, pero en realidad esto no es así. En los géneros classical y electronic hay una mejora del 2.37% y el 14.27% respectivamente en comparación con el experimento anterior; en cuanto al género metal/rock/punk los resultados obtenidos en ambos experimentos son idénticos; por último el género world empeora considerablemente ya que el porcentaje de aciertos se ha visto mermado un 10% si se compara con el experimento 1. Ante estos resultados cabe decir que este nuevo experimento no ha merecido la pena dado que la penalización que sufre el género world es mayor que las mejoras que se producen en los géneros classical y electronic.

Por último, la matriz de confusión de mejor caso (Tabla 15) presenta resultados similares a la mejor matriz de confusión del experimento 1 (Tabla 11). El género musical world es el que sigue dando más problemas puesto que presenta tasas de confusión extremadamente altas (un 43% de los archivos etiquetados como world es clasificado como metal/rock/punk). Por lo tanto el problema sigue sin resolverse cuando se usa una normalización por el máximo.

5.3 Experimento 3: Base de datos normalizada en media 0 y desviación típica 1

En este nuevo experimento el proceso a seguir es idéntico a los dos anteriores, la única diferencia es el método de normalización de la base datos: Ahora a cada característica se le resta la media de su grupo de características y se divide por la desviación típica de todas las características que componen su grupo.

Al igual que en los experimentos anteriores, las tablas Tabla 18, Tabla 19 y Tabla 20 muestran el porcentaje de aciertos por género para cada función de activación si se varían el número de neuronas de la capa oculta.

Nº nodos	**% Aciertos Classical**	**% Aciertos Electronic**	**% Aciertos Metal/Rock/Punk**	**% Aciertos World**	**Media de aciertos**
10	93,75	57,14	73,81	5,41	57,53
20	98,96	65,71	73,81	8,11	61,65
100	92,71	65,71	73,81	37,84	67,52
500	47,92	22,86	30,95	32,43	33,54

Tabla 18: Porcentaje de aciertos por género con función de activación sigmoidal y distinto número de nodos en la capa oculta.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	58,33	14,29	69,05	27,03	42,18
20	67,71	14,29	71,43	18,92	43,09
100	70,83	34,29	76,19	32,43	53,44
500	39,58	25,71	42,86	40,54	37,17

Tabla 19: Porcentaje de aciertos por género con función de activación sine y distinto número de nodos en la capa oculta.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	90,63	37,14	71,43	8,11	51,83
20	92,71	51,43	73,81	2,70	55,16
100	97,92	54,29	80,95	21,62	63,70
500	46,88	31,43	40,48	27,03	36,46

Tabla 20: Porcentaje de aciertos por género con función de activación hardlim y distinto número de nodos en la capa oculta.

Las siguientes tablas muestran, al igual que en los experimentos anteriores, las matrices de confusión para el mejor caso usando funciones de activación distintas.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	92,71%	1,04%	1,04%	5,21%
Electronic	11,43%	65,71%	11,43%	11,43%
Metal/Rock/Punk	0,00%	19,05%	73,81%	7,14%
World	21,62%	32,43%	8,11%	37,84%

Tabla 21: Matriz de confusión para 100 nodos en la capa oculta y función de activación sigmoidal.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	70,83%	10,42%	7,29%	11,46%
Electronic	28,57%	34,29%	25,71%	11,43%
Metal/Rock/Punk	11,90%	9,52%	76,19%	2,38%
World	27,03%	27,03%	13,51%	32,43%

Tabla 22: Matriz de confusión para 100 nodos en la capa oculta y función de activación sine.

	Classical	Electronic	Metal/Rock/Punk	World
Classical	97,92%	1,04%	0,00%	1,04%
Electronic	20,00%	54,29%	20,00%	5,71%
Metal/Rock/Punk	7,14%	11,90%	80,95%	0,00%
World	48,65%	21,62%	8,11%	21,62%

Tabla 23: Matriz de confusión para 100 nodos en la capa oculta y función de activación hardlim.

En este último experimento sí que se han conseguido mejorar los resultados. Ahora el porcentaje de aciertos global obtiene su máximo en 67.52% para la configuración de 100 nodos y función de activación sigmoidal. Estos resultados suponen una mejora sustancial con respecto a los dos experimentos anteriores que se nota sobre todo en los géneros musicales electronic y world. En el primero de ellos se ha conseguido ascender de un 45.71% de aciertos en el experimento 2 a un 65.71%, mientras que en el género world el ascenso ha sido desde un 18.91% que se lograba alcanzar en el experimento 1 a un 37,84%. Sin embargo el género metal/pop/rock ha sufrido un descenso notable del número de aciertos, bajando a un 73.81% de aciertos de este experimento desde el 90.47% que se conseguía en los géneros anteriores. Por último el género classical sigue cosechando buenos resultados ya que se ha conseguido obtener un 92.71% de aciertos. En términos generales compensa normalizar la base de datos en media y desviación típica pese a que el género metal/rock/pop ha empeorado ya que en los demás géneros se ha conseguido una mejora considerable.

5.4 Conclusiones

En este apartado se ha visto la evolución que sufre la clasificación de los cuatro géneros musicales propuestos si se va modificando la normalización de los datos, el número de nodos de la capa oculta y la función de activación empleada en la capa intermedia de la red. Según los resultados obtenidos estos tres parámetros influyen de forma notable en los resultados que ofrece el clasificador, obteniéndose los mejores resultados con una normalización en media cero y desviación típica uno en una red neuronal de 100 nodos en la capa oculta y función de activación sigmoidal.

Al normalizar la base de datos, en cualquiera de las dos normalizaciones, se ha observado que el número de nodos necesarios para obtener los mejores resultados se ha reducido (pasando de 500 nodos en el experimento 1 a 100 nodos en los experimentos 2 y 3). Esto supone una mejora en cuanto a tiempo de cálculo, por lo que en los próximos experimentos se va a trabajar solamente con la base de datos normalizada. Además como los mejores resultados se consiguen con la base de datos normalizada en media y varianza, está será la normalización que se emplee en los próximos capítulos.

A pesar de la mejora que se obtuvo en el último experimento con respecto a los dos anteriores, los resultados finales no son suficientes. La baja tasa de aciertos que presenta el género world es preocupante y se debe a que este género musical engloba a la música tradicional, música popular y étnica así como algún género específico de una zona concreta. Estos hechos hacen que los patrones que siguen este tipo de canciones sean muy distintos entre sí, por lo que es muy difícil su clasificación. Aun así se van a buscar soluciones que consigan mejorar los resultados obteniendo tasas de acierto aceptables para un sistema de clasificación automático.

6 Búsqueda de nuevas características

Dados los pobres resultados que se han obtenido en el capítulo anterior se ha decidido introducir mejoras con el fin de obtener unas tasas de acierto aceptables (de al menos el 70% de aciertos en todos los géneros musicales). Para ello primero se buscarán nuevas características que ofrezcan más información sobre cada género musical, de esta forma el clasificador tendrá más posibilidades de distinguir correctamente los cuatro géneros musicales propuestos. La información extra que se va conseguir viene proporcionada por unos coeficientes denominados Coeficientes Cepstrales de Mel, cuya extracción se explica con más detalle en el siguiente apartado.

Toda la teoría con la que ha sido posible elaborar este capítulo se ha obtenido de (Bonomo Laynez, 2012) y (Wikipedia).

6.1 Coeficientes Cepstrales de Mel

La extracción de los coeficientes cepstrales de Mel es una de las técnicas de extracción de parámetros más utilizadas e importantes en sistemas de reconocimiento de voz. Estos coeficientes son unos coeficientes particulares que vienen derivados de la aplicación del Cepstrum. El Cepstrum es un operador que transforma una convolución en el dominio temporal en una suma en el dominio frecuencial, consiguiendo separar la excitación y el tracto vocal de una señal de voz. La definición de Cepstrum se conoce como la transformada inversa de Fourier del logaritmo del espectro de la señal de voz, cuya expresión viene dada por la ecuación (15)

$$Cepstrum\,(s[n]) = \hat{s}[n] = F^{-1}[\log(|F(s[n])|] \tag{15}$$

Desarrollando un poco más la expresión anterior queda:

$$Cepstrum(s[n]) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\left(\left|S\left(e^{j\omega}\right)\right|\right) d\omega \tag{16}$$

Casi todas las técnicas de extracción de características vocales hacen uso del Cepstrum, sin embargo raramente es utilizado directamente debido a la influencia que tienen en él los efectos del canal. Es por ello que surgen los coeficientes cepstrales de Mel, los cuales

hacen uso de una escala no lineal denominada escala de Mel cuyo objetivo es imitar el comportamiento del oído humano asignando mayor resolución a las bajas frecuencias.

Los MFCC consiguen obtener una representación fuerte y compacta para obtener un modelo de la voz con un alto grado de precisión, es por eso que esta técnica ha sido considerada como una de las técnicas de parametrización de la voz más importantes. A continuación la Figura 3 muestra un esquema básico del proceso a seguir para obtener los MFCC:

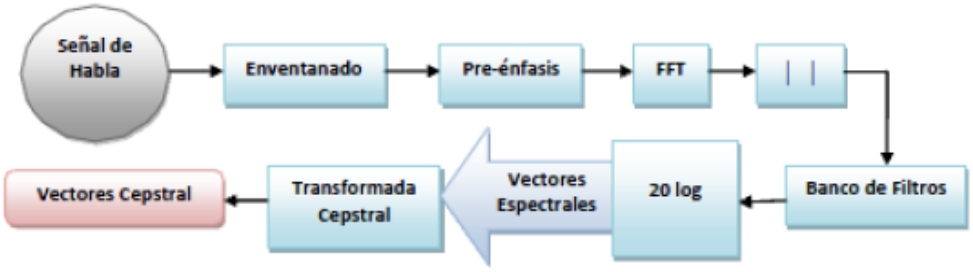

Figura 3 : Diagrama de bloques para la obtención de los MFCC. Figura adaptada de [Eugenio Arévalo, 2011].

Ahora se analizará con detalle cada uno de los bloques de la Figura 3:

- **Enventanado:** Debido a la naturaleza aleatoria y no estacionaria de la señal de voz, la obtención de características de la voz supone un gran inconveniente. Para solucionarlo se recurre a utilizar ventanas que generan tramas de muy corta duración, consiguiendo así que la señal que contiene la trama sea prácticamente estacionaria. Al igual que para las características espectrales obtenidas anteriormente, el tipo de ventana a emplear es del tipo Hamming.

- **Pre-énfasis**: La señal de audio pasa por un filtro de pre-énfasis con el objetivo de compensar la atenuación que se produce en el proceso del habla. Mediante la utilización de este tipo de filtro se consigue mantener constante el espectro en toda la banda reduciendo así las inestabilidades de cálculo asociadas con operaciones aritméticas de precisión finita. Este paso es opcional pero muy recomendable. La función de transferencia del filtro es la mostrada en la ecuación (17)

$$y[n] = x[n] - \alpha x[n-1] \tag{17}$$

  donde α suele tomar valores entre 0.95 y 0.98.

- **DFT:** Después de enventanar se calcula la DFT de N muestras a cada ventana utilizando la ecuación (18)

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk}, \;\; 0 \leq k \leq N. \tag{18}$$

A partir de aquí se trabaja con la envolvente de la señal de voz |X[k]|, descartando la componente de fase.

- **Banco de filtros:** El siguiente paso es multiplicar |X[k]| por un banco de filtros triangular equiespaciados según la escala de frecuencias de Mel.

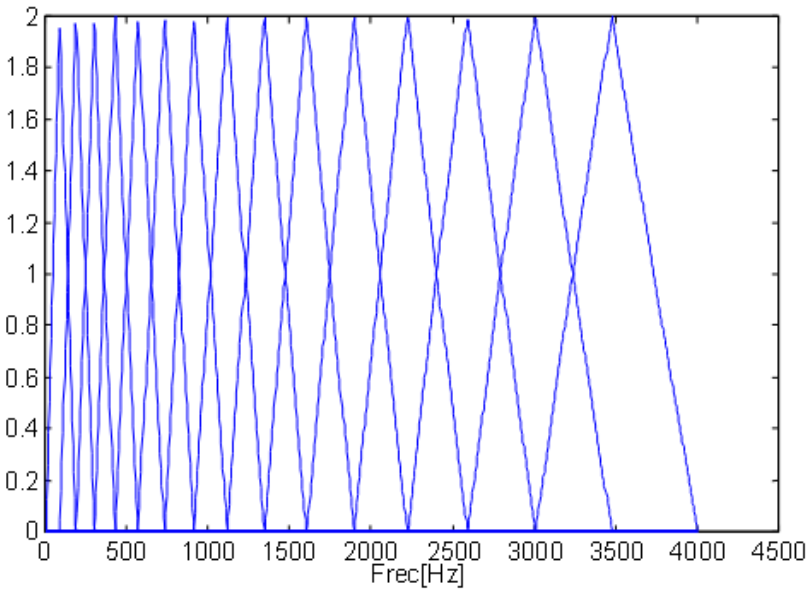

Figura 4: banco de filtros triangulares

Analizando la figura se ve como el banco de filtros está formado por 15 filtros triangulares equiespaciados en el dominio Mel. El ancho de bada de los filtros viene dado por la frecuencia central de cada filtro, la cual es función de la frecuencia de muestreo y del número de filtros. Con este filtrado se obtienen las bandas de energía de la señal que posteriormente van a ser tratadas para obtener los MFCC´S.

Por último se va a mostrar la relación entre la frecuencia y la escala de Mel. La ecuación (19) es la que proporciona esta relación.

$$mel(f) = 2595 \, log_{10}\left(1 + \frac{f}{700}\right) \tag{19}$$

- Después de multiplicar el espectro vocal con el banco de filtros, hay que calcular la energía correspondiente en cada uno de los F filtros implementados según la ecuación (20).

$$E_m = \sum_{k=0}^{N-1} |X[k]|^2 H_m[k], \;\; 1 \leq m \leq F \tag{20}$$

A continuación se calcula el logaritmo pasando por tanto al dominio de la potencia espectral logarítmica. Este dominio tiene el inconveniente de que los espectros de los filtros adyacentes tienen un alto grado de correlación, por lo que los coeficientes espectrales son estadísticamente dependientes entre ellos.

- Para eliminar esa dependencia se utiliza la Transformada Discreta del Coseno (DCT), que obtiene finalmente los coeficientes cepstrales buscados. La ecuación (21) muestra el procedimiento:

$$C_{MFCC}(m) = \sum_{k=0}^{N-1} \log(E_k) \cos\left(m\left(k - \frac{1}{2}\right)\frac{\pi}{N}\right), \;\; 1 \leq m \leq F \tag{21}$$

6.2 Implementación en Matlab de los MFCC

Para la obtención de estos coeficientes en Matlab se optó por buscar una librería ya implementada que agilizara el trabajo. La página (Slaney) contiene una serie de herramientas que implementan varios modelos auditivos populares para el entorno de programación Matlab, entre las cuales se encuentra la obtención de los MFCC. Dentro del conjunto de herramientas la función encargada de obtener los coeficientes cepstrales de Mel es la función mfcc.m. A continuación la Tabla 24 va a mostrar el prototipo de la función y sus parámetros más relevantes:

[ceps,freqresp,fb,fbrecon,freqrecon] =mfcc(input, samplingRate, [frameRate])	
Argumentos de entrada	
input	Señal audio muestreada a partir de la cual se van a obtener los coeficientes
samplingRate	Frecuencia de muestreo

frameRate	Longitud de la ventana a emplear
Argumentos de salida	
ceps	Coeficientes Cepstrales Mel
freqresp	El módulo de FFT empleada en el proceso de cálculo de los MFCC
fb	La escala de Mel a la salida del banco de filtros
fbrecon	La salida del banco de filtros basándose en la inversión de los coeficientes cepstrales con una transformada del coseno.
freqrecon	La respuesta en frecuencia lisa mediante la interpolación de la reconstrucción de fb. 256 canales para que coincida con la respuesta en frecuencia original.

Tabla 24: Prototipo de la función MFCC y sus correspondientes argumentos de entrada y de salida.

La ejecución de la función mfcc.m obtiene 13 Coeficientes Cepstrales de Mel los cuales se añadirán a la base de datos ya existente, para que finalmente cada archivo de audio contenga 23 características. Una vez que se ha actualizado toda la base de datos ya se pueden introducir los nuevos datos en el clasificador.

El siguiente capítulo analizará los resultados que consigue la ELM cuando se introduce en ella este pack nuevo de 23 características.

6.3 Resultados

Este apartado mostrará los resultados que ofrece la ELM multicanal empleada en los capítulos anteriores cuando se le introduce la base datos de 23 características (las 10 características del capítulo 3 más los 13 MFCC) normalizada en media cero y desviación típica uno. Esta normalización será la única que se utilice en estas nuevas pruebas debido a que ha sido la normalización que mejores resultados ha obtenido en los experimentos anteriores.

La función de activación utilizada en estas pruebas únicamente será la función sigmoidal dado que es la que mejores resultados obtuvo en el experimento 3 del

capítulo anterior. Se realizarán varias pruebas con distintos nodos en la capa oculta para ver cual es la que mejores tasas de acierto obtiene. La Tabla 25 es la encargada de representar los resultados.

Nº nodos	% Aciertos Classical	% Aciertos Electronic	% Aciertos Metal/Rock/Punk	% Aciertos World	Media de aciertos
10	95,833	48,571	61,905	13,514	54,956
20	92,708	62,857	73,810	10,811	60,046
100	95,833	71,429	83,333	43,243	73,460
500	62,500	25,714	50,000	37,838	44,013

Tabla 25: Porcentaje de aciertos para varios nodos en la capa oculta y función de activación sigmoidal.

Viendo los resultados proporcionados por la Tabla 25 se descubre que al igual que ocurría en el experimento 3 con la función de activación sigmoidal (Tabla 18), la mejor media global de aciertos se consigue con la configuración de 100 nodos en la capa oculta. Al utilizar esta configuración se consigue una media de aciertos en el global del 73.46%, lo que supone un aumento de 5.94 puntos con respecto a la mejor media global del experimento 3 (100 nodos y función de activación sigmoidal). Este aumento en el porcentaje global supone que los Coeficientes Cepstrales de Mel aportan información relevante para el clasificador. Aun así esta mejora no es suficiente dado que el género musical etiquetado como world sigue con un porcentaje de aciertos inferior al 50%. Pese a esta baja tasa de aciertos, con la inclusión de los MFCC en la base de datos se ha conseguido aumentar el porcentaje de aciertos de este género en un 14.03% con respecto a estudio homólogo realizado en el experimento 3. En cuanto a los otros 3 géneros musicales también han sufrido alguna mejora con respecto al experimento 3 en la configuración de 100 nodos y función de activación sigmoidal. Estas mejoras se van a detallar a continuación:

- El género classical es el que menos ha mejorado, solamente un 3.4%. La leve mejoría que ha sufrido este género se debe a los buenos resultados que obtuvo este género en el experimento anterior (92.71% de aciertos).

- El género electronic mejora en un 8.7%, esta es un avance considerable puesto que ahora el porcentaje de acierto de este género musical llega a una tasa de aciertos nada despreciable del 71.429%.

- El género metal/rock/punk sufre una mejora del 12.9% llegando hasta un porcentaje de aciertos del 83.3%.

Finalmente tras recoger todos los resultados se determina que esta solución no ha sido suficiente para alcanzar una calidad aceptable en la clasificación de géneros musicales debido a que el género musical world sigue lastrando los buenos resultados obtenidos en los restantes géneros musicales.

Dadas las circunstancias actuales, es necesario buscar nuevas soluciones que consigan clasificar de manera correcta los géneros musicales propuestos en este proyecto.

7 Modificación del Clasificador

Dados los malos datos que se han cosechado en el capítulo anterior al clasificar el género world, nace la necesidad de buscar soluciones que permitan mejorar ese porcentaje de aciertos. Este capítulo se centrará en encontrar una solución a ese problema, y para ello se buscará una solución muy simple: sustituir el clasificador multicanal por varios clasificadores binarios.

7.1 Clasificador Binario

Este clasificador es un caso particular de la ELM implementada hasta ahora. La estructura y el algoritmo de clasificación van a ser idénticos a los casos anteriores, lo único que cambia es que ahora en la salida de la red neuronal habrá dos nodos en vez de los cuatro que existían hasta ahora. De esta forma se simplifica notablemente el proceso de clasificación puesto que ahora cada clasificador nada más tendrá que distinguir entre dos posibles casos: si es el género musical buscado o por el contrario no lo es.

Para llevar a cabo este clasificador se han realizado modificaciones en la ELM implementada hasta este momento. A continuación se explicarán los cambios que se han realizado:

- En primer lugar, al utilizar clasificadores binarios es necesario utilizar más de uno. En este caso será preciso el uso de 3 clasificadores.

- Para "crear" estos tres clasificadores se harán 3 copias de las funciones elm_train.m y elm_predict.m que serán modificadas para albergar estos nuevos clasificadores.

- La modificación que hay que hacer a cada función es cambiar el nombre de la función y el nombre de los archivos donde la red neuronal almacena los datos relevantes. De esta forma cada red neuronal estará compuesta por un par de funciones elm_train_x.m y elm_predict_x.m, donde la x indica el número de clasificador.

- Por último hay que cambiar las etiquetas con la solución que tiene la base de datos existente en función del clasificador en el que se introduzcan los datos con

el objetivo de que el clasificador solamente detecte una posible solución: resultado válido o resultado no válido.

7.2 Resultados

Una vez hechos todos los cambios mencionados en el apartado anterior, es el momento de ver los resultados que se obtienen cuando se utiliza la red de ELM binarias al introducirles las 23 características obtenidas en los capítulos anteriores. Al igual que en capítulo anterior las pruebas se realizarán con la función de activación sigmoidal y la base de datos normalizada en media cero y desviación típica 1.

Al tener varios clasificadores se va a poder distinguir cuál es el género musical que se clasificará primero y cual el último. La lógica dice que en primer lugar deben clasificarse los géneros que han obtenido un mayor porcentaje de aciertos en los experimentos anteriores para dejar para el final el género musical que menor porcentaje de aciertos en los experimentos anteriores. Este hecho hace que estos resultados se compongan de varias pruebas en las que influirá el orden de clasificación de los géneros musicales así como el número de nodos que tendrá la capa oculta de cada ELM.

7.2.1 Prueba 1

En esta primera prueba la base de datos se va a dejar en su estado original: En un primer lugar el primer clasificador distinguirá entre música clásica y otros tipos de música. La segunda ELM será la encargada de averiguar si la música que se le introduce es electrónica o por el contrario pertenece a otro género musical. Por último el tercer clasificador será el que distinga entre metal/pop/rock y música del tipo world.

Con este orden en la base de datos se obtienen los siguientes resultados, reflejados en la Tabla 26

Nº nodos	**% Aciertos Classical**	**% Aciertos Electronic**	**% Aciertos Metal/Rock/Punk**	**% Aciertos World**	**Media de aciertos**
30/100/100	94,79	65,71	21,62	40,48	55,65
30/100/200	86,46	60,00	51,35	35,71	58,38
30/50/80	100,00	68,57	21,62	38,10	57,07
30/50/70	92,71	71,43	24,32	38,10	56,64

Tabla 26: Porcentaje de aciertos por género de la red ELM binaria en la primera prueba.

Los resultados que se ven en la tabla 26 no son a priori nada buenos puesto que se ha empeorado notablemente el porcentaje de aciertos en el género metal/rock/punk, pasando de un 83.3% que se obtuvo en el experimento anterior, a porcentajes de aciertos como mucho del 51.35%. Esto supone un retroceso como mínimo del 62.2%, lo que suponen que estos resultados son claramente inaceptables. Sin embargo los demás géneros musicales poseen una tasa de aciertos que no fluctúa en exceso con respecto al mejor resultado de las pruebas realizadas en el apartado anterior (Tabla 25). Para poder mejorar estos resultados se va a realizar una prueba nueva en la que se cambie el orden en el que los géneros musicales entran en los clasificadores.

7.2.2 Prueba 2

En esta nueva prueba para intentar levantar los malos resultados que ha obtenido el género metal/rock/punk se va a modificar el orden en el que los géneros musicales entren en el clasificador. Se intercambiarán dos géneros musicales entre sí: el género metal/rock/punk pasará al puesto de género classical y viceversa. El motivo de este cambio es facilitar la clasificación del género world del classical, puesto que el género classical es fácilmente reconocido por la ELM vistos los resultados que se han recogido hasta el momento.

La Tabla 27 muestra el porcentaje de aciertos por géneros que proporciona las ELM´s binarias cuando se le introduce el conjunto de características obtenido en capítulos anteriores.

Nº nodos	**% Aciertos Metal/Rock/Punk**	**% Aciertos Electronic**	**% Aciertos Classical**	**% Aciertos World**	**Media de aciertos**
30 50 70	60,42	54,29	72,97	72,97	65,16
50 50 70	58,33	54,29	72,97	72,97	64,64
30 80 70	57,29	62,86	72,97	72,97	66,52
30 50 70	60,42	65,71	67,57	67,57	65,32

Tabla 27: Porcentaje de aciertos por género de la red ELM binaria en la segunda prueba.

En esta nueva prueba se confirma que el orden en que se introduzcan los géneros musicales en el clasificador influye de forma notable en los resultados que se obtienen. Así en este nuevo experimento la media global no llega a alcanzar los buenos resultados que se obtuvieron en la Tabla 25, pero sí que se ha mejorado de forma notable el porcentaje de aciertos en el género world pasando ahora a un porcentaje de aciertos de

72.97%. Esto supone una mejora con respecto los porcentajes de acierto del género world que se recogen en la Tabla 25 de casi un 60%. Sin embargo los demás géneros musicales bajan el porcentaje de aciertos con lo que la media global queda 7 puntos por debajo de la mejora media recogida en la Tabla 25. Esta prueba incita a seguir probando variantes en lo que respecta al orden de introducción de los géneros musicales en los clasificadores, con el objetivo de mejorar los resultados actuales.

7.2.3 Prueba 3

En esta nueva prueba se va a probar una variante de la primera prueba: el primer clasificador debe catalogar solamente música clásica, el segundo debe hacer lo mismo con la música electrónica, y por último el tercer clasificador tiene encomendada la tarea de distinguir entre el género metal/rock/punk y world. A priori este ejemplo es igual que la prueba 1, pero en realidad no es así. Esta prueba cambiará el orden en que se introducen en el clasificador los datos, introduciendo en este caso en primera instancia en el tercer clasificador el género world y posteriormente el género metal/rock/punk. De esta forma se introducen en orden inverso los géneros musicales en el tercer clasificador.

Nº nodos	**% Aciertos Classical**	**% Aciertos Electronic**	**% Aciertos World**	**% Aciertos Metal/Rock/Punk**	**Media de aciertos**
30 70 50	93,75	74,29	78,38	83,33	82,44
30 40 50	95,83	62,86	72,97	92,86	81,13
30 100 50	90,63	74,29	78,38	80,95	81,06
30 50 30	91,67	71,43	81,08	88,10	83,07
30 80 30	96,88	71,43	83,78	85,71	84,45
30 50 50	93,75	77,14	91,89	88,10	87,72

Tabla 28: Porcentaje de aciertos por género de la red ELM binaria en la tercera prueba.

Los resultados proporcionados por esta prueba son extremadamente buenos. En la configuración de 30, 50 y 50 nodos se consigue una media de global del 87.72% de aciertos, valor impensable de alcanzar hasta ahora. Este hecho hace que esta

configuración sea la configuración definitiva que adquiera el proyecto puesto que bate con soltura todos los registros que se habían recogido hasta el momento.

A continuación los resultados recogidos en la Tabla 28 para la configuración de 30, 50 y 50 nodos se van a comparar con los resultados de la configuración de 100 nodos recogidos en la Tabla 25 puesto que esos eran los mejores resultados obtenidos en este TFG hasta ahora. La razón por la que han sido catalogados como los mejores resultados es que esa configuración conseguía la media global de aciertos más alta, a pesar de que el género world no obtuviese una buena tasa de aciertos. la Tabla 29 será la encargada de mostrar los resultados de las dos mejores configuraciones y la mejora que se consigue en la última prueba para género musical.

Nº nodos	**% Aciertos Classical**	**% Aciertos Electronic**	**% Aciertos Metal/Rock/Punk**	**% Aciertos World**	**Media de aciertos**
100	95,83	71,43	83,33	43,24	73,46
30 50 50	93,75	77,14	88,10	91.89	87,72
	Mejora por género				
	-2.17%	8.00%	5.72%	112.50%	19.41%

Tabla 29: Mejores resultados del TFG y la mejora que se ha conseguido en la última prueba.

Observando la Tabla 29, se aprecia que todos los géneros musicales sufren una mejora a excepción del genero classical que empeora en un 2.17%. Este empeoramiento no supone un inconveniente debido a que aun así la tasa de aciertos de este género musical se sitúa por encima del 90%. En cuanto a los géneros electronic y metal/rock/punk sufren una leva mejoría haciendo aún mejores los resultados obtenidos en la configuración de 100 nodos. Sin embargo donde más se nota el incremento del porcentaje de aciertos es en el género world. Con la nueva configuración que se ha probado en este último experimento se ha logrado conseguir que este género musical aumente su tasa de aciertos en un 112.5%, llegando así a una excelente tasa de aciertos que alcanza el 91.89%. Gracias a esta última mejora se logra que la media global de aciertos alcance el 87.72%, cifra que ofrece una calidad muy buena para el sistema implementado.

En consecuencia, estas nuevas mejoras han llevado a obtener los resultados que se esperaban conseguir de un sistema de estas características.

8 Validación de los resultados

Este nuevo capítulo servirá para verificar los buenos resultados que se han cosechado en el capítulo anterior. Para ello se hará uso de la segunda parte de la base de datos, la parte de validación. Se extraerán las 23 características obtenidas en las secciones anteriores para toda la base de datos, se normalizarán todas esas características en media cero y desviación típica uno, y finalmente, se introducirá todo ese paquete de información en la red de binaria de ELMs. La estructura que va a seguir este título se descompone en dos apartados: un primer apartado en el que se presentarán los resultados que obtenga el clasificador de forma ideal, y un segundo apartado en el que se muestren los datos simulando una situación real.

8.1 Simulación de resultados de forma ideal

Este apartado mostrará los porcentajes de aciertos que se consiguen a la salida de la red binaria de clasificadores cuando se introducen los datos en la red de forma "ideal". Este comportamiento ideal se debe a que las canciones de la base de datos se eliminan según su clase para introducirlas en los clasificadores. Por ejemplo, en el primer clasificador se introduce la base de datos entera distinguiendo entre música clásica de la que no lo es. A continuación en el segundo clasificador entrará toda la base de datos menos las canciones de música clásica, y finalmente en el último clasificador entrarán las canciones correspondientes a los dos últimos géneros musicales. En este tipo de clasificación se están eliminando las canciones de datos según su clase real, mientras que en la simulación de forma real los datos se eliminan según los resultados que proporcionen los clasificadores. Hasta ahora los datos recopilados en la red binaria de clasificadores se habían conseguido según el procedimiento de la forma ideal, por lo que este apartado servirá para confirmar si los resultados recogidos en el capítulo anterior son válidos o no.

La Tabla 30 muestra los resultados que se han obtenido para la base de datos de validación cuando la red está entrenada con la configuración de 30 50 y 50 nodos y función de activación sigmoidal, que era el mejor caso que se había conseguido en todo el trabajo.

% Aciertos Classical	% Aciertos Electronic	% Aciertos World	% Aciertos Metal/Rock/Punk	Media de aciertos
95,00	75,44	82,79	85,03	84,56

Tabla 30: Resultados obtenidos para la base de datos de validación en el caso ideal.

Como se puede comprobar, los datos recogidos mantienen la buena tendencia que se venía recogiendo hasta el momento, lo que significa que el clasificador musical funciona correctamente. Se observa que la tasa de acierto se mantiene constante con variaciones menores del 10% en todos los géneros musicales. El género que mayor variación sufre es el género world debido a la gran diversidad de música que hay englobada en este género musical, lo que hace que sea muy complicado averiguar todas las canciones que conforman este género. Aun así la clasificación de este género musical es muy buena puesto que la tasa de aciertos que maneja este género está por encima del 80%.

8.2 Simulación de resultados de forma real

Ahora se va a modificar el modo en que se introducen los archivos de entrada en los clasificadores, eliminando las canciones de la base de datos según los resultados de salida del clasificador. Por ejemplo, en el primer clasificador entra toda la base de datos, las canciones que este primer clasificador detecte como clásicas serán desechadas de la base de datos. Las canciones restantes pasarán al segundo clasificador, el cual detectará las canciones cuyo género musical es electrónico. Acto seguido se eliminaran de la base de datos las canciones que se hayan detectado como electrónicas, Para que al siguiente clasificador solamente pasen las canciones restantes. Este proceso se repite de forma iterativa.

En la Tabla 31 se muestran los porcentajes de acierto que se obtienen a la salida de la red binaria de clasificadores cuando se introducen los datos según el criterio anterior.

% Aciertos Classical	% Aciertos Electronic	% Aciertos World	% Aciertos Metal/Rock/Punk	Media de aciertos
95,00	64,81	63,92	90,32	78,51

Tabla 31: Resultados obtenidos para la base de datos de validación en el caso real.

Como se puede ver en la tabla, en este caso ha empeorado bastante el porcentaje de aciertos de los géneros musicales world y electronic. ¿A qué se debe este empeoramiento? Básicamente se debe a que el o los clasificadores anteriores clasifican

erróneamente canciones de otros géneros que al ser eliminadas de la base de datos no pasan a su correspondiente clasificador haciendo que baje la tasa de aciertos. Por ejemplo, una canción de género electronic. Esta canción en la simulación ideal ha sido catalogada como clásica pero no se elimina de la base de datos porque en este tipo de simulación solamente se eliminan las canciones cuya etiqueta real es clásica, por lo tanto la canción pasa al segundo clasificador. Este nuevo clasificador analiza la canción y detecta que efectivamente esa canción pertenece al género electronic. Sin embargo si se aplica la simulación del caso real, el primer clasificador habría detectado la canción como clásica y acto seguido se hubiese eliminado de la base de datos. Este acto hace que la canción no llegue al segundo clasificador, por lo que la canción no se puede detectar como electrónica haciendo que disminuya la tasa de aciertos del género electronic.

Aun así la tasa global de aciertos se eleva hasta el 78.51% gracias al alto porcentaje de aciertos que hay en los géneros classical y metal/rock/punk.

9 Interfaz Gráfica

Con el objetivo de hacer el proyecto más amigable de cara al público, se ha tomado la decisión de dotar al proyecto de una interfaz gráfica con la que los usuarios puedan interactuar sin hacer uso de la línea de comandos.

Para implementar la interfaz gráfica se ha hecho uso de la herramienta GUIDE que proporciona Matlab.

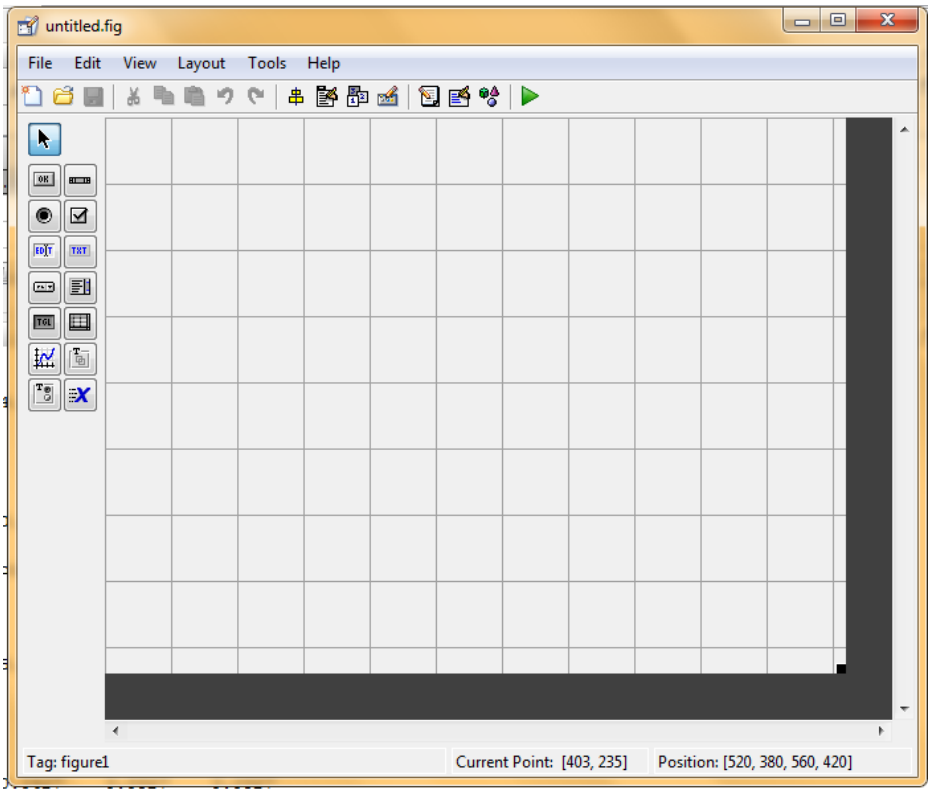

Figura 5: Imagen de la herramienta de trabajo GUIDE.

Como se puede apreciar en la Figura 5, la herramienta GUIDE está compuesta por un área central denominada área de diseño, una zona lateral izquierda conocida como paleta de componentes y una zona superior en la que se encuentran varias herramientas. En el área de diseño se van colocando componentes de la paleta de componentes hasta que el programa adquiera el resultado visual deseado. Se pueden insertar botones, cuadros de texto, menús despegables, gráficas, chexbox, etc.

El funcionamiento de una aplicación GUI en Matlab consta de dos archivos básicos: uno con extensión .m y otro con extensión .fig. El archivo .fig contiene los elementos

gráficos, mientras que el archivo .m contiene el código con la correspondencia de los botones de control de la interfaz gráfica. Al añadir un elemento nuevo en la interfaz gráfica el archivo .m se actualiza automáticamente generando código para ese nuevo elemento.

La utilidad que se le quiere dar a la interfaz gráfica es la clasificación de archivos de audio por parte de un usuario común. Para ello se ha pensado en que el programa tenga un cuadro de texto en el que se introduzca el nombre de la canción a clasificar y un par de botones para pasar la canción a formato numérico y clasificarla. Además se va a colocar una gráfica en el que se muestre la señal de audio. El resultado final sería el siguiente:

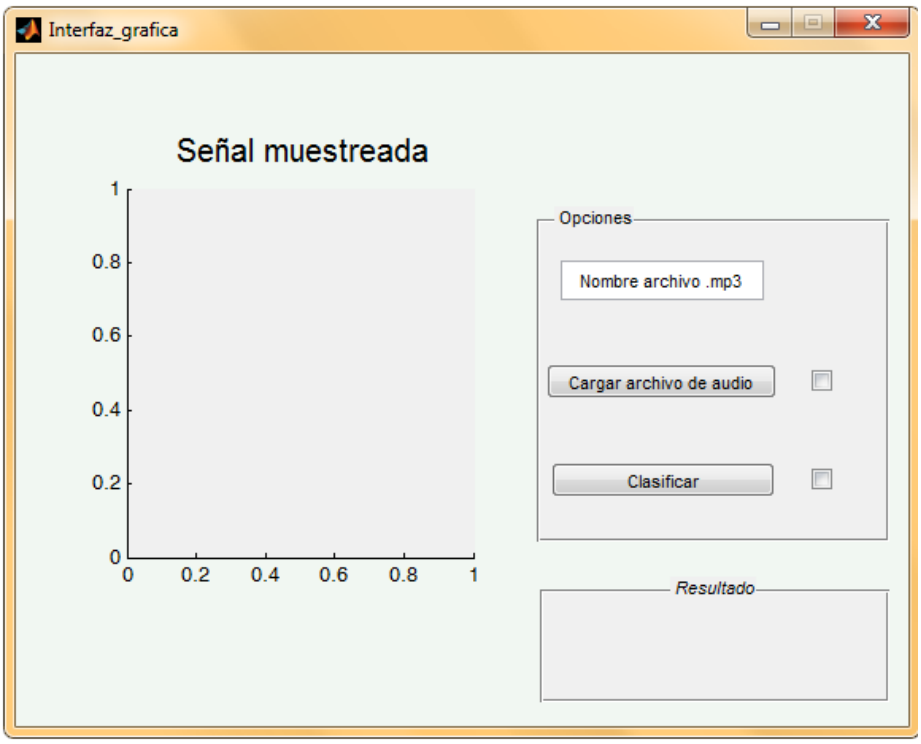

Figura 6: Aspecto final de la interfaz gráfica.

Observando la Figura 6, se puede ver como la interfaz gráfica diseñada está compuesta por un cuadro de texto en el que se introduce el nombre del archivo a analizar, un gráfico en el que se verá la señal muestreada, dos botones y un cuadro de texto para el resultado de la clasificación. El funcionamiento de la interfaz es el siguiente: primero se introduce el nombre del fichero de audio que se desee analizar, acto seguido el usuario deberá pulsar el botón de "cargar archivo de audio" para convertir el archivo .mp3 en una matriz numérica que será representada en gráfico de la izquierda. Una vez que se ha realizado correctamente la carga de archivo se mostrará un tick en el checkbox que la derecha del botón, indicando que el proceso de carga ha finalizado. Cuando finaliza el proceso de carga se puede pasar a la clasificación del archivo, para ello se pulsará el

botón clasificar. Cuando el programa termine de clasificar se mostrará un tick en el checkbox correspondiente y aparecerá el resultado final el cuadro habilitado para ello. Para cargar un nuevo archivo basta con introducir de nuevo el nombre en cuadro de texto y pulsar intro. En ese instante se eliminará el resultado de la canción anterior y se borrarán los ticks de los checkbox.

Por último la Figura 7 va a mostrar el comportamiento real de la interfaz gráfica diseñada.

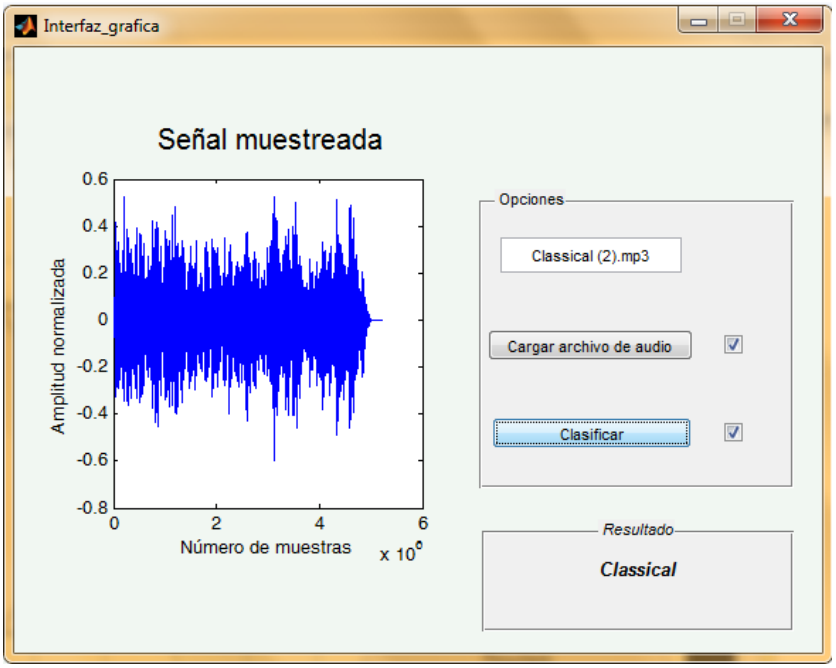

Figura 7: Funcionamiento real de la interfaz gráfica.

Como se puede ver en la imagen, el archivo de audio está cargado correctamente y clasificado, por lo que se muestran dos ticks en los checkbox. Además en la gráfica se ha dibujado la señal de audio, con lo que el usuario puede ver la amplitud de la señal que va a clasificar.

10 Conclusiones

Este trabajo fin de grado ha tenido el objetivo de resolver de una forma rápida y eficaz el lento proceso, que se venía haciendo de forma manual, del etiquetado de géneros musicales. Para conseguirlo se han seguido una serie de pasos fundamentales:

- Se buscó una base de datos ya etiquetada que sirviese como referencia para trabajar sobre ella. De esta forma se consigue saber si los resultados que se obtienen son correctos o no.

- Sobre la base de datos descargada se aplicaron técnicas de procesado de señal cuyo objetivo era obtener información esencial sobre los archivos de cada género musical, para posteriormente analizar esos datos y poder seguir patrones de clasificación.

- Con el conjunto de características ya extraído, se implementó una red neuronal artificial encargada de crear los patrones necesarios para la correcta clasificación de los géneros musicales propuestos.

- Dados los malos resultados que se obtuvieron, se decidió buscar soluciones que intentaran resolver los problemas que se presentaron. Finalmente esas soluciones consistieron en obtener más características fundamentales y en modificar el clasificador montando una serie de clasificadores binarios en cadena.

- Con todo esto finalmente se consiguió obtener unos resultados excelentes situando la tasa media de aciertos en torno al 87%.

Durante todo este proceso se han tenido que tomar decisiones muy importantes que han influido en los resultados finales del trabajo. En primer lugar la base de datos estaba muy descompensada, con un gran número de archivos para algún género y muy pocos archivos para otros. La solución que se adopto fue eliminar algún género musical y agrupar otros con el fin de que más o menos todos los géneros musicales contaran con un número de archivos similar. Otro dilema que surgió a la hora de implementar el clasificador es si había que normalizar los parámetros que se introducían en la ELM o por el contrario no. En caso de tener que normalizarlos, ¿cuál sería la mejor normalización? Con la ayuda de varias pruebas al final se comprobó que los mejores resultados se obtuvieron siempre con la características de la base de datos normalizadas en media cero y desviación típica 1, por lo que a partir de ese momento fue la

normalización que se empleó. Por otra parte, de entre las funciones de activación que había disponibles en la ELM se ha optado por usar la función sigmoidal por dos razones: ante el mismo número de nodos los resultados que proporcionaba esta función eran mejores que los del resto, y además es más eficiente en cuanto a tiempo que tarda en entrenar la red neuronal.

Al validar los resultados que se obtuvieron en el desarrollo se detectó que simulando el comportamiento real de la red los resultados de dos géneros musicales empeoraron. Ese inconveniente se contrarresta con el ahorro de tiempo y dinero que conlleva clasificar de forma manual tantas canciones.

Finalmente se optó por proporcionar una interfaz gráfica al proyecto. Esta interfaz facilita el uso del clasificador musical al usuario común ya que no es necesario que haga uso de la ventana de comandos para hacer llamadas a funciones e interpretar resultados.

10.1 Líneas futuras

- Mejorar la interfaz gráfica: La interfaz gráfica que dispone el proyecto es realmente simple, por eso se puede mejorar introduciendo mejoras con el GUI de Matlab o se puede intentar programar en otros lenguajes de programación.

- Utilización de otros algoritmos de clasificación: Se puede optar por buscar otro tipo de redes, o por implementar modificaciones de la ELM básica como por ejemplo la OP-ELM viendo si los resultados mejoran o por el contrario, empeoran.

- Búsqueda de nuevas características. Hasta ahora las características que se han extraído pertenecen a características musicales relacionadas con el timbre. Se pueden buscar otro tipo de características relacionadas con el ritmo, la melodía, etc.

- Ampliar el número de géneros musicales a clasificar. Hasta ahora solamente han sido cuatro los géneros musicales que se han clasificado, pero se puede intentar mejorar el proyecto buscando nuevas bases de datos que incluyan más géneros musicales.

- Utilizar la base del proyecto para investigar sobre la identificación vocal, tipos de instrumentos musicales en un fragmento de audio, etc.

Pliego de condiciones

En este nuevo capítulo del proyecto se incluyen las condiciones bajo las que se ha desarrollado el presente trabajo fin de grado.

Condiciones de materiales y equipos

Hardware utilizado

Se ha utilizado un ordenador portátil con las siguientes características para la creación y ejecución de los algoritmos que se usan en el proyecto, así como la edición de textos e imágenes.

- Microprocesador Intel Pentium ® Dual-Core P7450 @2.13 GHz.
- Memoria RAM de 4 GB.
- Disco Duro de 250 GB.
- Tarjeta gráfica de 1 GB.
- Periféricos: ratón y teclado (incluido en el portátil).

Software utilizado

- Sistema operativo:
  Windows 7 Home Premium 64 bits.
- Software de desarrollo:
  MATLAB Rb2014a.
- Procesador de textos:
  Microsoft Word 2010.
- Procesador de imágenes:
  Microsoft Paint.
- Procesador de cálculo:
  Microsoft Excel.

Conexiones de red

Para la búsqueda y descarga de artículos ha sido necesaria una conexión a la red privada de la UAH con la que se han podido descargar gratuitamente los artículos necesarios para la realización del proyecto.

Condiciones de ejecución

Gracias a la interfaz gráfica que se ha diseñado cualquier usuario que disponga de Matlab puede ejecutar el programa de clasificación musical, aunque no tenga conocimientos de programación. Sin embargo si se quiere entrenar de nuevo las redes neuronales cambiando el número de nodos y funciones de activación o introduciendo nuevas bases de datos, es necesario que el usuario tenga unos conocimientos mínimos de informática puesto que hay que trabajar con diversas funciones.

Presupuesto

Este último título presenta los costes asociados a la realización de este proyecto. Se realizará un desglose de este presupuesto con el fin de aportar mayor facilidad a la comprensión los gastos.

Coste del material informático

En este apartado se desglosan los costes relacionados con los recursos necesarios para la realización y edición de documentos de este proyecto.

Concepto	**Total**
Ordenador portátil con Windows 7	550 €
Impresora Epson Bx525WD	120 €
Software Matlab R2014a	2.000 €
Microsoft Office 2010	117,60 €
Paint	0
TOTAL	**2787,60 €**

Tabla 32: Coste del material utilizado en el proyecto.

Costes de personal

A continuación se detallarán los salarios de los empleados necesarios para la realización del proyecto.

Este proyecto ha necesitado la contratación de un ingeniero técnico de telecomunicaciones encargado del desarrollo del proyecto. Ha necesitado unas 480 horas (6 horas diarias, 5 días a la semana durante 4 meses). Mientas que para la documentación se ha contratado a un administrativo encargado de documentar el proyecto, éste ha necesitado aproximadamente unas 100 horas.

Concepto	Sueldo (€/hora)	Horas trabajadas	Total
Ingeniero	40	480	19.200 €
Administrativo	15	120	1.800 €
TOTAL			**21.000 €**

Tabla 33: Coste de personal

Costes extras

En este nuevo apartado se reflejarán costes extraordinarios que han sido necesarios para la elaboración del trabajo fin de grado. Aquí se recogen los costes asociados a la encuadernación e impresión del proyecto, así como diversos costes de oficina (folios, bolígrafos, grapadoras,...). La impresión del proyecto se ha realizado con la impresora que se documentó en los costes de material informático, por lo que los costes de impresión que se reflejan en la siguiente tabla corresponden a la compra de folios y cartuchos de tinta.

Concepto	Total
Impresión	18 €
Encuadernación	90 €
Material de oficina	10 €
TOTAL	**118 €**

Tabla 34: Costes extras.

Presupuesto final

Por último se presenta el coste final del proyecto, que resulta de sumar todos los costes anteriores y aplicarles el 21% de IVA. Como los gastos informáticos llevan el IVA incluido, no es necesario volvérselo a aplicar. La Tabla 35 muestra el coste asociado al aplicar el IVA.

Concepto	% IVA	Total
IVA material informático	Ya incluido	0 €
IVA coste personal	21	4.410 €
IVA costes extras	21	37,80 €
TOTAL		**4.447,80 €**

Tabla 35: Coste del IVA.

Finalmente la Tabla 36 muestra el precio final del proyecto.

Concepto	Total
Costes totales	23.908,60 €
IVA	4.447,80 €
TOTAL	**28.353,40 €**

Tabla 36: Presupuesto final.

El presupuesto final de este proyecto asciende a la cantidad de **veintiocho mil trescientos cincuenta y tres y cuarenta céntimos** (28.353,40 €).

En Alcalá de Henares, a 18 de septiembre de 2014

Fdo.: Diego López Pajares
Graduado en ingeniería en tecnologías de telecomunicación.

Bibliografía

Alvarado, D. (18 de Abril de 2005). *Efecto del enventanadoen la obtención del espectro discreto de una señal.* Obtenido de Monografías.com: http://www.monografias.com/trabajos20/enventanado/enventanado.shtml

Bonomo Laynez, D. (2012). Extracción de Características. En D. Bonomo Laynez, *Sistemas de verificación automática de locutor.* Universidad de Sevilla.

Crespo, A. B. (2013). *Aprendizaje Máquina Multitarea mediante Edición de Datos y Algoritmos de Aprendizaje Extremo.* Universidad Politécnica de Cartagena.

Fernandez, A. (31 de Octubre de 2004). *Mathworks.* Obtenido de http://www.mathworks.com/matlabcentral/fileexchange/6152-mp3write-and-mp3read

García Laencina, P. J., Verdú Monedero, R., Larrey Ruiz, J., Morales Sánchez, J., & Sancho Gómez, J. L. (2010). *Nuevas Tendencias en Redes Neuronales Artificiales: Extreme Learning Machine.* Universidad Politécnica de Cartagena. Escuela Técnica Superior de Ingeniería de Telecomunicación.

Guangbin , H., Quin-Yu, Z., & Chee-Kheong , S. (2006). Extreme learning machine: Theory and aplications. *Neurocomputing.*

Guangbin, H. (s.f.). *Extreme Learning Machine.* Obtenido de http://www.ntu.edu.sg/home/egbhuang/

Guangbin, H. (s.f.). *Extreme Learning Machine.* Obtenido de http://www.ntu.edu.sg/home/egbhuang/source_codes/elm_train_predict.zip

Guangbin, H., Hongming, Z., Xiaojian, D., & Rui, Z. (2012). Extreme learning machine for regression and multiclass classification. *IEEE transactions on systems, man and cybernetics. Vol. 42, No. 2.*

MIREX. (2005). *Music Information Retrieval Evaluation eXchange.* Obtenido de http://www.music-ir.org/mirex/wiki/2005:Audio_Genre

Music Technology Group. (2004). *Universitat Pompeu Fabra.* Obtenido de http://www.iua.upf.edu/mtg/ismir2004/contest/Training_Tracks1.tar.gz

http://www.iua.upf.edu/mtg/ismir2004/contest/Training_Tracks2.tar.gz
http://www.iua.upf.edu/mtg/ismir2004/contest/Development_Tracks1.tar.gz
http://www.iua.upf.edu/mtg/ismir2004/contest/Development_Tracks2.tar.gz

Nam, U. (28 de Abril de 2001). *Special Area Exam Part II.* Obtenido de Center for Computer Research in Music and Acoustics, Stanford University: https://ccrma.stanford.edu/~unjung/AIR/areaExam.pdf

Slaney, M. (s.f.). *Auditory Toolbox.* Obtenido de https://engineering.purdue.edu/~malcolm/interval/1998-010/

Tzanetakis, G., & Cook, P. (2002). Automatic Musical Genre Classification. *IEEE transactions on speech and audio processing, Vol. 10,Nº5.*

Wikipedia. (s.f.). *Wikipedia.* Obtenido de http://en.wikipedia.org/wiki/Mel-frequency_cepstrum